Linux Básico

Quinta clase



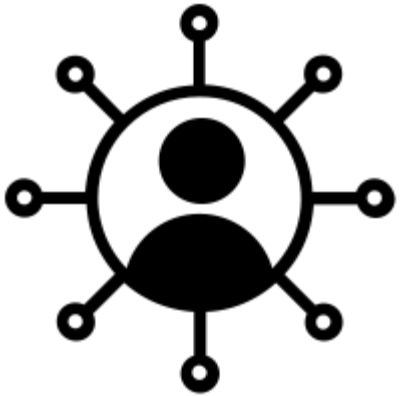
Redes y conectividad

NetworkManager

Utiliza el concepto de **perfiles de conección**

→ Estos contienen la configuración de la red

- Int32
- Boolean
- Uint32
- String
- Array of string
- uint64



Utiliza **plugins** para leer y escribir los datos con el formato de archivo **INI-key**

```
[connection]
id=net_connection
uuid= 9cd8a444-f501-4179-900e-f3754dbbe7c0
type=ethernet
autoconnect=true

[ipv4]
method=auto

[ipv6]
method=auto

[ethernet]
mac-address=48:2a:e3:8f:4b:aa
```

- /etc/NetworkManager/system-connections/
- /usr/lib/NetworkManager/ system-connections/
- /run/NetworkManager/system-connections/

NetworkManager command-line interface (nmcli)

NetworkManager puede manejarse desde línea de comandos mediante nmcli, nos permite:

- Crear
- Mostrar
- Editar
- Borrar → **conexiones de red**
- Activar
- Desactivar

Así mismo, se puede **controlar** y **mostrar** el estado de los **dispositivos de red**

Puede ejecutarse como un **comando** desde el terminal o ser invocado desde un **script**

- Como **comando** que se ejecuta desde el terminal, nmcli puede crear, editar, iniciar y detener conexiones de red o mostrar el estado de un dispositivo de red, sin necesidad de la GUI o el applet de escritorio.
- ```
  nmcli [OPTIONS] OBJECT { COMMAND | help }
  ```

- Para los **scripts**, nmcli proporciona un formato de salida sencillo que puede adaptarse al procesamiento de datos, que integra las configuraciones de red en un proceso en lugar de gestionar las conexiones de red manualmente.
  ```

Manos a la obra

Para obtener una visión general de las configuraciones de red de los dispositivos conectados

Ejecutar los siguientes comando:

```
nmcli -o conn
```

```
nmcli -o dev show <device>
```

```
nmcli -f TYPE,FILENAME,NAME conn
```

Configuración manual de una dirección IP

1. Guardar un respaldo de la configuración actual

```
cat /run/NetworkManager/system-connections/Wired\
connection\ 1.nmconnection >> wired_connection_1.bkp.txt
```

2. Crear un nuevo perfil

```
nmcli connection add con-name wired-conn1
      ifname enp1s0 type ethernet
```

3. Asignar una dirección IP al perfil creado

```
nmcli connection modify wired-conn1 ipv4.addresses
192.168.122.100/24
```

4. Asignar la configuración de IP como manual

```
nmcli connection modify wired-conn1 ipv4.method manual
```

Hasta aquí ya hemos configurado la dirección IP de nuestro sistema...

...Aún no es suficiente, son necesarias otras configuraciones para usar nuestra red.

Configurar el default gateway

```
nmcli connection modify wired-conn1 ipv4.gateway 192.168.122.1
```

Configurar el dns

```
nmcli connection modify wired-conn1 ipv4.dns
192.168.122.1
```

Activar el perfil de conección

```
nmcli connection up wired-conn1
```

Ver el estado de los dispositivos

```
nmcli device status
```

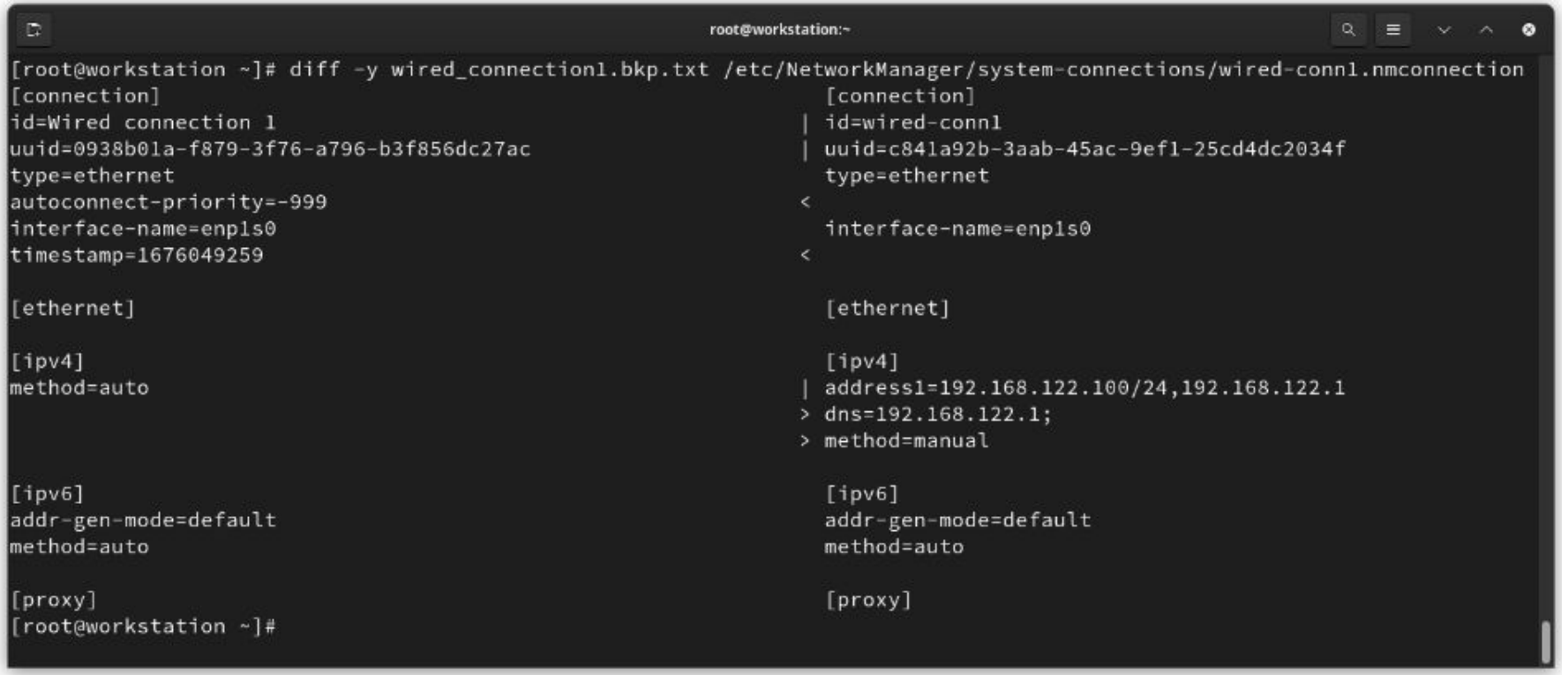
Revisar la configuración de red del entorno de trabajo

```
ip a
```

Revisar los keyfiles del entorno de trabajo

```
nmcli -f TYPE,FILENAME,NAME conn
```

Parámetros obtenidos:



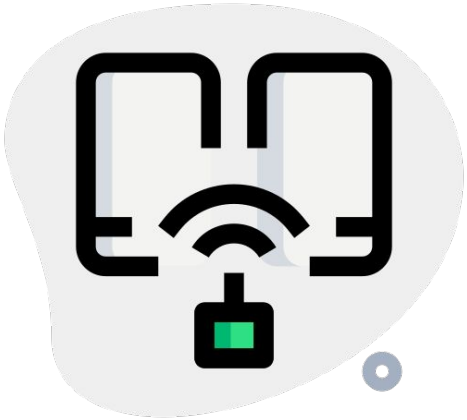
Ajuste de la conectividad inalámbrica

Verificar que sea compatible con nuestro SO a través del reconocimiento de su **chipset**

- PCI: `lspci -nn`
- USB: `lsusb:`

Una de las principales características a revisar de los anteriores comandos son los parámetros:

- Vendor ID (VID)
- Product ID (PID).



Linux Kernel Driver DataBase (https://cateee.net/lkddb/) -> **"<chipset>" site:cateee.net/lkddb/**

Se presentará una gran cantidad de sitios útiles, seleccionar el relacionado con el hardware de linux en https://linux-hardware.org/

Para identificar el dispositivo:

1. Identificar el comando lspci para ver la nic

```
lspci | grep -i broadcom
```

2. Obtener información detallada con base en lo anterior

```
lspci -vv -s 00:00.0 [bus]
```

3. Enumera los módulos que se cargaron en el kernel y busca el chipset identificado de la tarjeta inalámbrica

```
lsmod | grep -i broadcom
```

4. Mostrar la información del módulo

```
modinfo broadcom
```

Encontrar la mejor calidad de conexión a la red

Mostrar las posibles conecciones wi-fi (access points)

```
nmcli dev wifi list
```

- ➔ Cian señal débil (menos del 30% de intensidad).
- ➔ Magenta señal más fuerte (30-50%)
- ➔ Naranja indica una señal mejor (60-80%)
- ➔ Verde representa una señal excelente (80-100%).

Establecer una conección:

```
nmcli device wifi connect [SSID] password
[SSID-password]
```

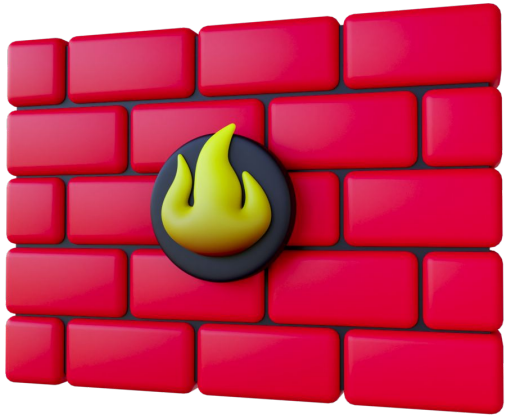
Monitorear constantemente el estado deldispositivo de red:

```
nmcli device show <Wlan0>
```

¿Y la seguridad?

El principio más básico de la seguridad es ser consciente de lo que tenemos abierto, como **puertos**, **sockets**, **archivos** y **servicios/procesos**

- La herramienta que nos permite monitorear aquellas conexiones que deseamos mantener es el **cortafuegos.**

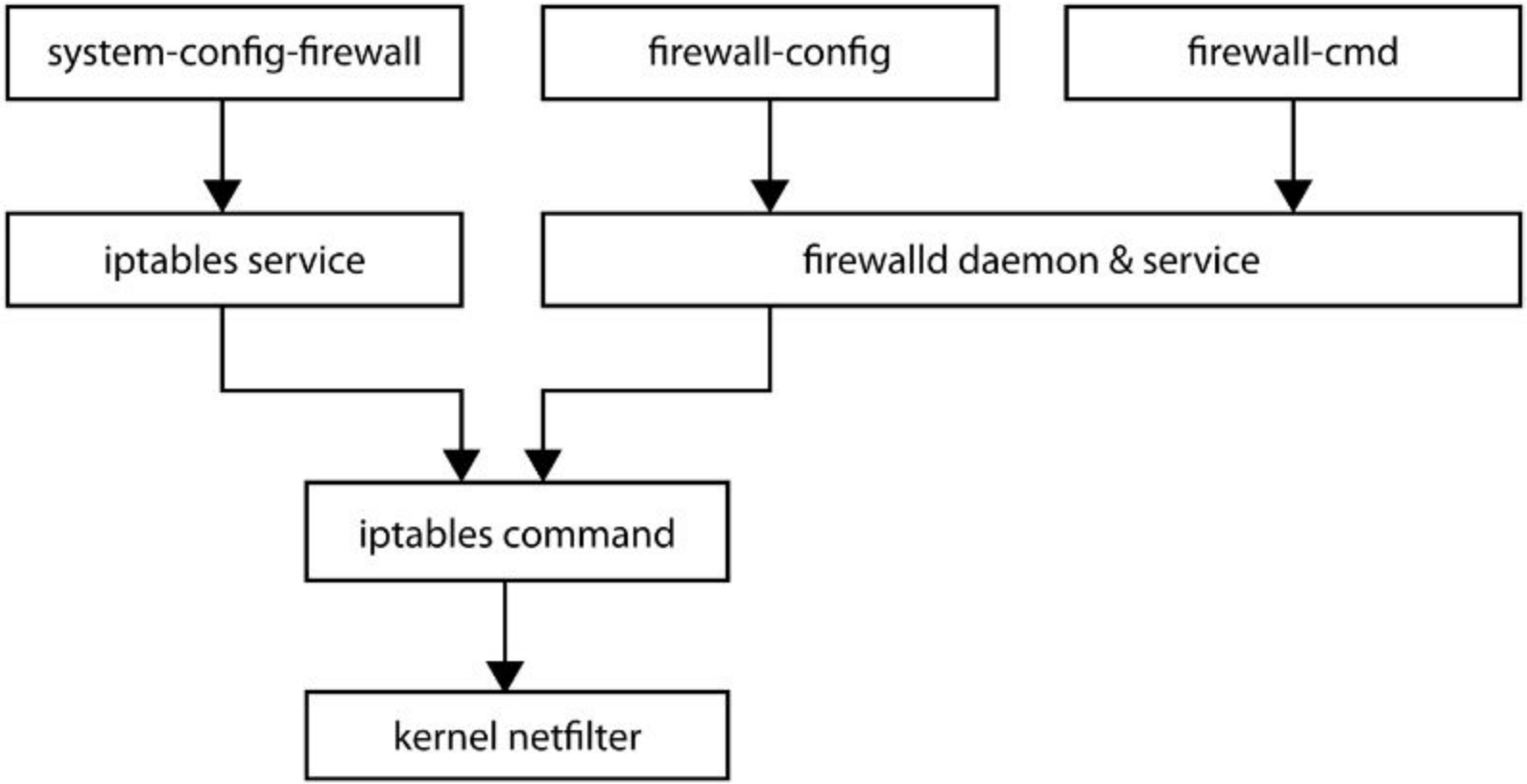


Como su nombre indica, un cortafuegos de red es una **barrera protectora** que impide las comunicaciones de red no autorizadas. Su principal objetivo es impedir el acceso o uso no autorizado de los servicios de red de nuestro sistema.

Fedora Linux tiene un cortafuegos incorporado como parte de las funciones de red dentro del kernel. El servicio de gestión en Fedora utiliza **firewalld** provee administración dinámica de cortafuegos con soporte para zonas de red/cortafuegos.

Este soporte define el nivel de confianza de las conexiones o interfaces de red. La interfaz D-Bus de firewalld utiliza las herramientas de configuración firewall-cmd, firewall-config y firewall-applet **herramientas de configuración de cortafuegos**.

El siguiente diagrama muestra el flujo de gestión del cortafuegos con firewalld.


system-config-firewall
firewall-config
firewall-cmd
iptables service
firewalld daemon & service
iptables command
kernel netfilter

Por defecto, el servicio firewalld viene instalado con Fedora. Puede utilizar la interfaz **firewall-cmd** cli para investigar su estado.

```
firewall-cmd --state
```

Para cargar la configuración en el cortafuegos:

```
firewall-cmd --list-all
```

En nuestro caso de uso de estación de trabajo, es probable que no necesitemos tener servicios o puertos expuestos a la red. Por lo tanto, la mejor práctica es cerrarlos todos. Antes de cerrarlos, utilice el comando ss para investigar los puertos que han sido abiertos por los procesos para determinar si se refieren a los servicios y puertos permitidos en el cortafuegos:

```
ss -tulpn
```

Con lo anterior se debe hacer un análisis inteligente de los servicios realmente empleados por nosotros y cerrar aquellos que no utilicemos
Para eliminar definitivamente los servicios de la configuración, utilice el comando firewall-cmd

```
firewall-cmd --permanent --delete-service={dhcpv6-client,mdns,samba-client}
```

Para eliminar los puertos dinámicos

```
firewall-cmd --permanent --remove-port={1025-65535/udp,1025-65535/tcp}
```

Establecer y verificar los cambios

```
firewall-cmd --reload
```