1. Introducción. Puertos de red

En la gestión de redes y operaciones de servidores, comprender con precisión el estado de los puertos es esencial. Especialmente al usar Ubuntu, verificar qué puertos están abiertos y qué procesos los están utilizando ayuda a reforzar la seguridad y permite una solución de problemas más rápida.

Este artículo explica los comandos y herramientas básicos para comprobar puertos en Ubuntu. Proporciona pasos prácticos y fáciles de entender para usuarios principiantes e intermedios, así que asegúrate de leer hasta el final.

2. ¿Qué es un puerto?

2.1 Concepto básico de los puertos

Un puerto es un punto final de comunicación virtual que las computadoras y dispositivos de red utilizan para enviar y recibir datos. Específicamente, cuando varias aplicaciones se comunican simultáneamente en la misma dirección IP, los puertos identifican y enrutan los datos hacia la aplicación correcta.

Por ejemplo, un servidor web usa el puerto 80 para tráfico HTTP. Si el mismo servidor permite acceso SSH, utiliza el puerto 22. Dado que los servicios se distinguen por números de puerto, comprobar el estado de los puertos es crucial en la gestión de redes.

2.2 Tipos y funciones de los puertos

Los puertos se clasifican en tres grupos principales:

1. **Puertos bien conocidos (0–1023)**

- Números de puerto estandarizados globalmente asignados a servicios de uso frecuente.
    - Ejemplos:
        - HTTP: 80
        - HTTPS: 443
        - SSH: 22

2. **Puertos registrados (1024–49151)**

- Puertos utilizados por aplicaciones o empresas específicas. wp:list /wp:list
    - Ejemplos:
        - MySQL: 3306
        - PostgreSQL: 5432

3. **Puertos dinámicos (49152–65535)**

- Puertos usados temporalmente por aplicaciones, comúnmente en comunicaciones del lado del cliente.

Comprender esta clasificación facilita determinar para qué se utiliza cada número de puerto.

3. Cómo comprobar puertos en Ubuntu

Ubuntu ofrece varias herramientas para verificar el estado de los puertos. Esta sección explica cuatro comandos particularmente útiles.

3.1 Uso del comando ss

El comando `ss` es una herramienta potente de gestión de redes para sistemas Linux. Se ejecuta rápidamente y proporciona información detallada de las conexiones.

Comando básico:

```
sudo ss -ltn
```

Detalles de las opciones:

- **-l** : Muestra solo los puertos en estado LISTEN.
- **-t** : Muestra solo el protocolo TCP.
- **-n** : Muestra direcciones y números de puerto en forma numérica.

Salida de ejemplo:

```
State       Recv-Q Send-Q      Local Address:Port        Peer Address:Port
LISTEN      0      128              0.0.0.0:22               0.0.0.0:*
```

3.2 Uso del comando netstat

El comando `netstat` ha sido ampliamente utilizado como herramienta de gestión de redes durante muchos años. Aunque está siendo reemplazado gradualmente por `ss`, sigue estando disponible en muchos sistemas.

Comando básico:

```
sudo netstat -ltn
```

Salida de ejemplo:

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
```

3.3 Uso del comando lsof

`lsof` es útil para identificar los procesos que están usando puertos específicos.

Comprobar un puerto concreto:

```
sudo lsof -i :80
```

Salida de ejemplo:

```
COMMAND   PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
apache2   1234  www    4u   IPv4  12345 0t0      TCP *:http (LISTEN)
```

3.4 Uso del comando nmap

`nmap` es una herramienta de escaneo de redes que se usa frecuentemente para diagnósticos de seguridad.

Escanear localhost:

```
sudo nmap localhost
```

Salida de ejemplo:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-21 18:00 JST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00013s latency).
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
```

Puntos clave:

- Se listan los puertos abiertos y los servicios asociados.
- Es posible escanear servidores externos, pero se requiere la autorización adecuada.

4. Comprobación de la configuración del firewall

En Ubuntu, los firewalls se utilizan comúnmente para mejorar la seguridad. `ufw` (Uncomplicated Firewall), en particular, es una herramienta de gestión simple pero potente ampliamente utilizada para este propósito. Esta sección explica cómo verificar el estado de los puertos y modificar configuraciones usando `ufw`.

4.1 Verificar el estado de ufw

Comando para verificar el estado del firewall:

```
sudo ufw status verbose
```

Salida de ejemplo:

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
```

Explicación:

- **Status: active** — indica que el firewall está habilitado.
- **Logging: on** — el registro está habilitado y la actividad del firewall se está grabando.
- **Default: deny (incoming), allow (outgoing)** — las conexiones entrantes se deniegan por defecto, mientras que las conexiones salientes se permiten.
- **ALLOW** — muestra puertos o servicios que están explícitamente permitidos (por ejemplo, SSH y HTTP).

Consejo:
Si el firewall está deshabilitado (`Status: inactive`), habilítelo con el siguiente comando:

```
sudo ufw enable
```

4.2 Permitir o bloquear puertos

Comando para permitir un puerto:

```
sudo ufw allow 22/tcp
```

Explicación:

- Permite conexiones TCP en el puerto 22 (SSH).

Comando para bloquear un puerto:

```
sudo ufw deny 80/tcp
```

Explicación:

- Bloquea el acceso al puerto 80 (HTTP).

Ejemplo: Permitir acceso desde una dirección IP específica únicamente

```
sudo ufw allow from 192.168.1.100 to any port 22 proto tcp
```

Explicación:

- Permite conexiones SSH desde la dirección IP `192.168.1.100` únicamente.

4.3 Restablecer y revisar configuraciones

Para restablecer la configuración del firewall y comenzar de nuevo, ejecute el siguiente comando:

```
sudo ufw reset
```

Esto elimina todas las reglas y devuelve el firewall a su estado predeterminado. Si restablece las configuraciones, asegúrese de revisar y volver a aplicar las reglas necesarias.

5. Ejemplo práctico: Verificar el estado de un puerto específico

Esta sección proporciona un ejemplo práctico usando SSH (puerto 22) para mostrar cómo verificar el estado del puerto.

5.1 Verificar el estado del puerto

Comando de ejemplo:

```
sudo ss -ltn | grep 22
```

Salida de ejemplo:

```
LISTEN      0      128        0.0.0.0:22            0.0.0.0:*
```

Puntos clave:

- Si aparece `LISTEN` en la salida, el puerto está abierto y esperando conexiones.
- `0.0.0.0` indica que se aceptan conexiones desde todas las direcciones IP.

5.2 Verificar el proceso en ejecución

Comando de ejemplo:

```
sudo lsof -i :22
```

Salida de ejemplo:

```
COMMAND   PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd      1234  root   3u   IPv4  56789 0t0      TCP *:ssh (LISTEN)
```

Puntos clave:

- `sshd` es el proceso daemon que gestiona las conexiones SSH.
- Puede detener o reiniciar el proceso usando su ID de proceso (PID).

Ejemplo de detención de un proceso:

```
sudo kill 1234
```

5.3 Ejemplo de resolución de problemas

Problema: Qué hacer cuando un puerto está cerrado o inaccesible.

Pasos:

1. Verificar las configuraciones del firewall.

   ```
   sudo ufw status verbose
   ```

2. Si el puerto está bloqueado, permítalo.

   ```
   sudo ufw allow 22/tcp
   ```

3. Verificar el estado del servicio y reiniciarlo si es necesario.

   ```
   sudo systemctl restart ssh
   ```

6. Gestión de puertos y seguridad

La gestión de puertos está directamente relacionada con la seguridad de la red. Esta sección explica puntos clave sobre la apertura y cierre de puertos y el mantenimiento de configuraciones seguras.

6.1 Cerrar puertos no utilizados

Los puertos que no se utilizan deben cerrarse para reducir el riesgo de acceso no autorizado.

Ejemplo: Cerrar el puerto 80

```
sudo ufw deny 80/tcp
```

6.2 Contramedidas contra el escaneo de puertos

El escaneo de puertos es una técnica que los atacantes utilizan para identificar vulnerabilidades en un sistema. Los siguientes métodos ayudan a proteger su servidor:

1. **Fortalecer las reglas del firewall:**

   ```
   sudo ufw default deny incoming
   ```

2. **Monitorear registros:**

   ```
   sudo tail -f /var/log/ufw.log
   ```

3. **Instalar herramientas de detección de escaneo de puertos:** Use herramientas como `fail2ban` para bloquear automáticamente los intentos de acceso no autorizados.

7. Resumen

Este artículo explicó métodos y comandos específicos para comprobar puertos en Ubuntu. También cubrió la gestión del firewall usando `ufw` y medidas de seguridad prácticas.

7.1 Puntos clave

- **Conceptos básicos y categorías de puertos:** Los puertos sirven como puntos de entrada de comunicación y se clasifican en puertos bien conocidos, registrados y dinámicos.
- **Cómo comprobar puertos:** Comandos como `ss`, `netstat`, `lsof` y `nmap` proporcionan información sobre el estado de puertos y procesos.
- **Gestión del firewall:** Usando `ufw`, puedes permitir o bloquear puertos para mejorar la seguridad del sistema.
- **Importancia de la seguridad:** Cerrar puertos no utilizados, monitorear registros y usar herramientas de seguridad ayuda a mantener un entorno de red seguro.

7.2 Aplicación práctica

La gestión de puertos es un aspecto fundamental de la seguridad de red. Aplica los conocimientos adquiridos en este artículo para mantener un entorno de servidor seguro y estable.

Preguntas frecuentes: Preguntas frecuentes sobre la comprobación de puertos en Ubuntu

P1. ¿Qué debo hacer si un puerto no está abierto en Ubuntu?

R:
Intenta los siguientes pasos:

1. **Revisar la configuración del firewall:**

```
sudo ufw status verbose
```

Si el puerto está bloqueado, permítelo con:

```
sudo ufw allow [portnumber]/tcp
```

2. **Verificar que el servicio esté en ejecución:**

```
sudo systemctl status [servicename]
```

Ejemplo para SSH:
`sudo systemctl status ssh`

Reinicia si es necesario:

```
sudo systemctl restart [servicename]
```

3. **Confirmar que el puerto correcto esté configurado:** Revisa el archivo de configuración del servicio, como `/etc/ssh/sshd_config` para SSH, para verificar el número de puerto apropiado.

P2. ¿Cuál es la diferencia entre `ss` y `netstat`?

R:
Ambas herramientas se usan para comprobar conexiones de red, pero difieren de la siguiente manera:

- **`ss`:** La herramienta recomendada para sistemas Linux modernos. Es más rápida y brinda información más detallada. Ejemplo: `sudo ss -ltn`
- **`netstat`:** Una herramienta más antigua, que se está volviendo gradualmente obsoleta, pero aún está ampliamente disponible en sistemas heredados. Ejemplo: `sudo netstat -ltn`

Para sistemas más recientes, se recomienda `ss`.

P3. ¿Cómo puedo detectar escaneos de puertos?

R:
Utiliza los siguientes métodos:

1. **Revisar los registros del firewall:**

```
sudo tail -f /var/log/ufw.log
```

Busca direcciones IP sospechosas o intentos de acceso repetidos.

2. **Instalar herramientas IDS/IPS:**

- Usa herramientas como `fail2ban` o `Snort` para bloquear automáticamente los intentos de acceso no autorizados.

3. **Escanear tu propio servidor con nmap:**

```
sudo nmap localhost
```

Identifica puertos abiertos innecesarios y ciérralos.

P4. ¿Cómo puedo comprobar qué proceso está usando un puerto específico?

R:
Usa el comando `lsof`:

```
sudo lsof -i :[portnumber]
```

Ejemplo para el puerto 80:

```
sudo lsof -i :80
```

Salida de ejemplo:

```
COMMAND   PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
apache2   1234  www    4u   IPv4  12345 0t0      TCP *:http (LISTEN)
```

P5. ¿Cómo permito solo una dirección IP específica usando `ufw`?

R:
Ejecuta el siguiente comando:

```
sudo ufw allow from [IP address] to any port [portnumber] proto tcp
```

Ejemplo: Permitir acceso SSH desde `192.168.1.100`:

```
sudo ufw allow from 192.168.1.100 to any port 22 proto tcp
```

P6. ¿Cómo puedo cambiar el número de puerto?

R:
Edita el archivo de configuración del servicio correspondiente.
Ejemplo para SSH:

1. Edita el archivo de configuración:

   ```
   sudo nano /etc/ssh/sshd_config
   ```

2. Busca la directiva `Port` y establece un nuevo número de puerto:

   ```
   Port 2222
   ```

3. Reinicia el servicio SSH:

   ```
   sudo systemctl restart ssh
   ```

4. Permite el nuevo puerto a través del firewall:

   ```
   sudo ufw allow 2222/tcp
   ```

P7. ¿Puedo permitir varios puertos a la vez?

R:
Sí, puedes permitir varios puertos usando los siguientes métodos:

1. Permitiendo un rango de puertos:

   ```
   sudo ufw allow 1000:2000/tcp
   ```

Explicación: Permite puertos del 1000 al 2000.

2. Permitiendo puertos individualmente:

   ```
   sudo ufw allow 22/tcp
   ```