**ANEXO I**

**Práctica 1: Hardening.**

**Informe**

* **Documentar las configuraciones aplicadas.**

Las configuraciones realizadas se encuentran en la parte procedimiento de la práctica.

* **Presentar las capturas de pantalla, con la debida explicación de los resultados mostrados. Además, indicar el impacto en la seguridad.**

Los resultados obtenidos de la práctica con la debida explicación se muestran en la parte de resultados de la práctica. Y también el impacto que tiene en la seguridad de los dispositivos.

* **Generar un par de claves pública y privada utilizando ssh-keygen de manera que se tenga acceso al servidor solo mediante claves SSH.**

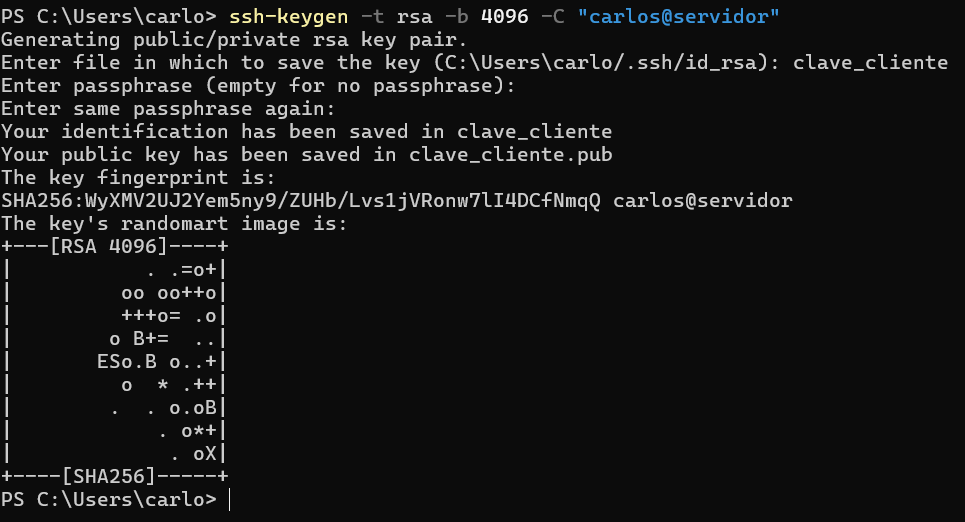
Para el caso se usa la máquina física como cliente y la máquina virtual (dispositivo linux) como servidor. De esta forma primero se debe generar el par de claves para acceder desde SHH, con el comando ssh-keygen -t rsa -b 4096 - -C "usuario@servidor".

Donde:

-t rsa → Especifica que se usará el algoritmo RSA.

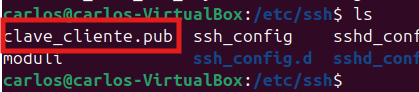
-b 4096 → Genera una clave de 4096 bits para mayor seguridad.

-C "usuario@servidor" → Agrega un comentario identificativo en la clave pública.



**Figura 5.1.** Generación de par de claves del cliente.

De esta forma se crean los archivos clave\_cliente, el cual es la clave privada y clave\_cliente.pub que es la clave pública, después se debe copiar el archivo clave\_cliente.pub al servidor



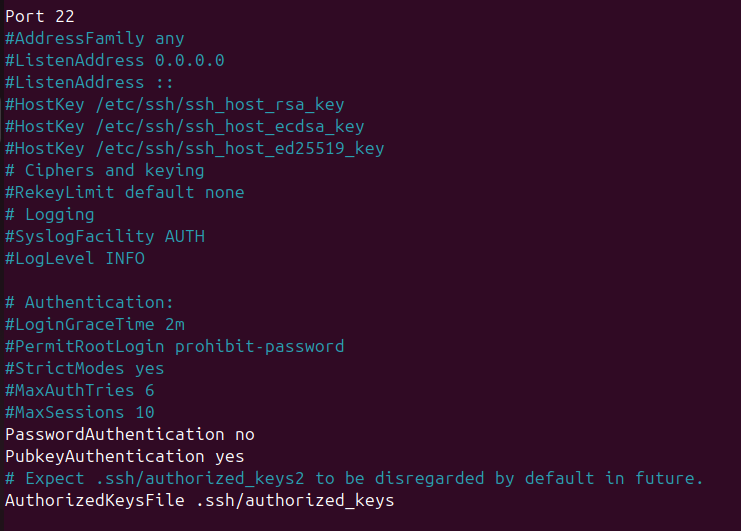
**Figura 5.2.** Copia de la clave publica del cliente al servidor

Una vez que el archivo clave\_cliente.pub se encuentra en el servidor se procede a configurar el archivo sshd\_config con el comando sudo nano /etc/ssh/sshd\_config para solo permitir claves SSH. La configuración se muestra a continuación:

PasswordAuthentication no → Deshabilita la autenticación por contraseña (solo claves SSH permitidas).

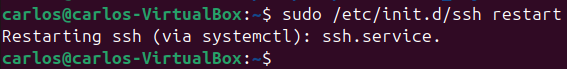
PubkeyAuthentication yes → Habilita la autenticación con claves públicas.

AuthorizedKeysFile .ssh/authorized\_keys → Especifica dónde están almacenadas las claves permitidas.



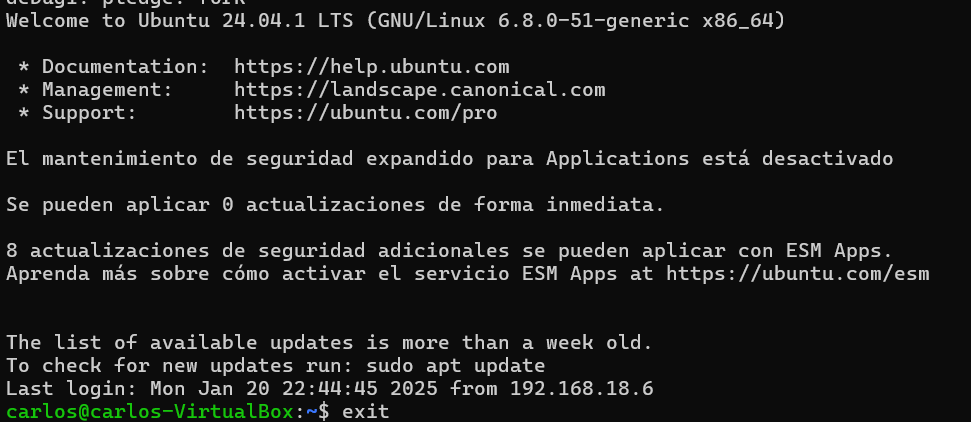
**Figura 5.3.** Configuración del archivo sshd\_config

Para este caso se habilita el puerto 22 (por defecto) para probar la conexión. Además, se crea el archivo authorized\_keys donde se copia el archivo la clave del archivo clave\_cliente.pub. Por último, se reinicia el servicio con el comando sudo /etc/init.d/ssh restart.



**Figura 5.4.** Reinicio del servicio SSH.

Después se prueba la conexión desde el terminal de Windows con el siguiente comando ssh -i $env:USERPROFILE\.ssh\clave\_cliente.pub. P



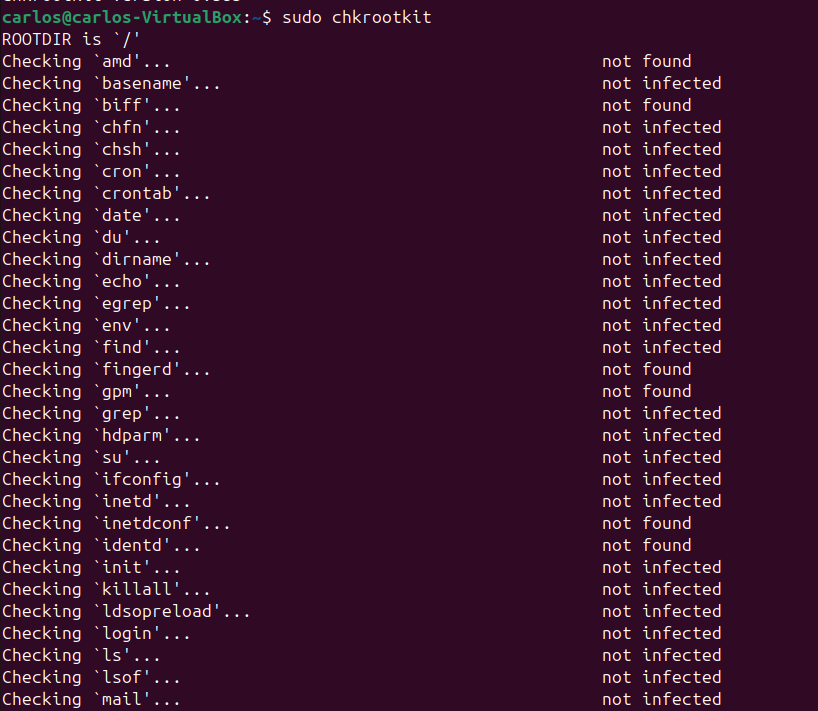
**Figura 5.5.** Prueba de conexión.

De esta forma se configuro para que únicamente la autenticación esté basada en claves, eliminando la autenticación por contraseña y mejorando la seguridad del servidor

* **Instalar y ejecutar chkrootkit para verificar la presencia de rootkits, además documentar y analizar los hallazgos.**

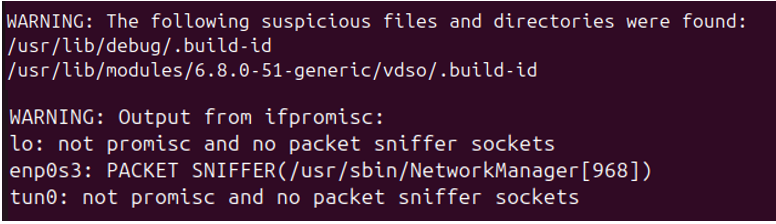
chkrootkit es una herramienta de código abierto que permite detectar rootkits en sistemas Linux. Un rootkit es un tipo de malware que otorga acceso privilegiado a un atacante, permitiéndole ocultar su presencia y persistir en el sistema.

De esta forma para instalar chkrootkit con el comando sudo apt install chkrootkit -y, luego se comprueba la instalación y la versión con el comando chkrootkit -V



**Figura 5.6.** Instalación de chrootkit.

También se puede usar el siguiente comando para guardar el archivo en un log y poder analizarlor, para ello se ejecuta el siguiente comando: sudo chkrootkit > /var/log/chkrootkit.log. De esta forma se tiene lo siguiente



**Figura 5.7.** Advertencias presentes en el sistema.

Se muestra las siguientes advertencias, por lo cual el sistema puede estar infectado por este tipo de malware. Por lo que se recomienda usar un antivirus para proteger el dispositivo de las posibles amenazas.

Por otro lado, para ver si existe conexiones activas y posibles comunicaciones maliciosas se usas el siguiente comando sudo netstat -tulnp | grep LISTEN



**Figura 5.8.** Conexiones maliciosas

Como se puede ver en la figura 5.8. no existen conexiones maliciosas. Además, también es posible hacer una lista de procesos activos y busca actividades sospechosas, para ello se usa el comando ps aux --sort=-%cpu | head -10



**Figura 5.9.** Procesos activos en el cpu.

Como se observa esta no se tiene actividades sospechosas dado que los procesos fueron ejecutados por el usuario carlos. Y son procesos ejecutados por el usuario.

De esta forma chkrootkit es una herramienta efectiva para detectar rootkits en sistemas Linux, la cual se complementa con rkhunter analiza el proceso sospechoso. De modo que lo más seguro es reinstalar el sistema desde una copia de seguridad limpia cuando en rootkit se confirme en el sistema.