Mecanismos de Acceso Remoto

Administración de Sistemas y Redes

2º Trabajo de Teoría

Pablo José Pérez Díez

UO282440

1. Introducción

En el contexto de la administración de sistemas y redes, el acceso remoto entre máquinas es una herramienta fundamental para la gestión eficiente y segura de infraestructuras informáticas. Este trabajo tiene como objetivo explorar distintos mecanismos de acceso remoto, evaluando su funcionamiento, características, ventajas y limitaciones de cada uno de ellos.

Este trabajo se dividirá en tres partes, en función de los sistemas operativos involucrados en cada sección.

En la primera parte se utilizarán técnicas de acceso remoto desde una máquina Linux a otra máquina Linux. A su vez, en la segunda sección se utilizarán técnicas distintas a las anteriores para llevar a cabo el acceso remoto desde una máquina Linux a una máquina Windows. Y, por último, veremos cómo se podría realizar lo mismo desde una máquina Windows a otra máquina Windows, utilizando técnicas diferentes a las utilizadas previamente.

En el entorno Linux a Linux, se utilizará SSH, como solución estándar para acceso remoto a través de terminal. Se mencionarán también otras opciones como el X11 Forwarding, el uso de servidores RDP en Linux y VNC para el acceso gráfico al entorno de escritorio como alternativas para realizar la conexión remota.

Para el acceso desde Linux a Windows, se empleará el protocolo RDP, aprovechando su soporte nativo en sistemas Windows. Se comentará también la posibilidad de utilizar WinRM para la administración remota.

Finalmente, en el entorno Windows a Windows, se utilizará PowerShell Remoting como solución principal. También se mencionarán herramientas como TeamViewer, que, aunque no formen parte del sistema operativo, son ampliamente utilizadas por su simplicidad y gran cantidad de funcionalidades.

Este trabajo será realizado utilizando diversas máquinas virtuales en un entorno local para probar todos los mecanismos de acceso remoto anteriormente mencionados.

2. Acceso Remoto de Linux a Linux

2.1 SSH

SSH (Secure Shell) es un protocolo de red utilizado para lograr una conexión segura entre dos máquinas, incluso a través de una red no segura, como Internet. Fue desarrollado como una alternativa segura a protocolos inseguros como Telnet y rlogin, que transmitían información en texto plano, lo que suponía un fallo grave de seguridad.

SSH permite a los usuarios acceder de forma remota a otro equipo para ejecutar comandos, transferir archivos o administrar servicios.

El funcionamiento de SSH se basa en la criptografía de clave pública y clave privada.

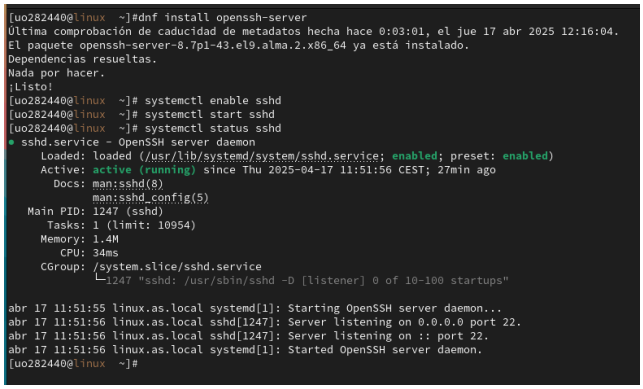
El proceso sería algo así:

El dispositivo que inicia la conexión solicita acceso al servidor SSH. Después, se lleva a cabo un intercambio de claves criptográficas que permite establecer una conexión cifrada. Este intercambio asegura que los datos enviados entre ambos dispositivos estén protegidos.

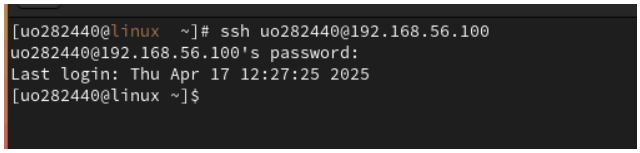
Tras establecer la conexión el servidor tiene varias maneras de confirmar la identidad del usuario, mediante contraseña o mediante clave pública. Una vez autenticado, toda la comunicación se cifra, asegurando la privacidad e integridad de los datos.

Ahora pasaremos a ver cómo se puede realizar una conexión mediante SSH entre dos máquinas Linux.

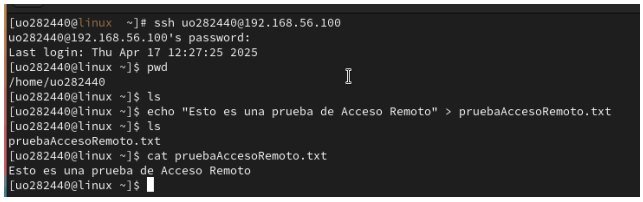
El primer paso sería instalar e iniciar el servicio SSH:



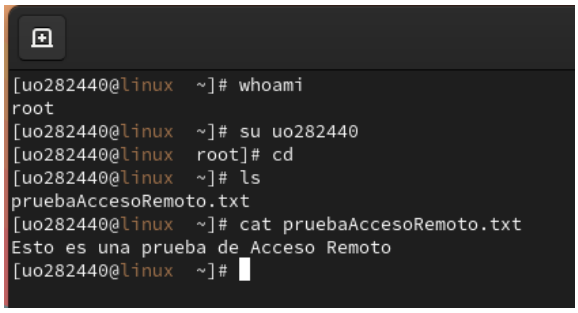
Cómo se puede ver “openssh-server” ya estaba instalado y solo tuvimos que iniciar el servicio. A continuación, procedemos a conectarnos al servidor:



Para comprobar que nos hemos conectado y todo funciona correctamente, creamos un archivo desde el cliente y lo visualizaremos en el servidor:



Y comprobamos que se muestra adecuadamente desde el servidor:



2.2 Alternativas

Cuando se busca establecer una conexión remota entre dos máquinas Linux, SSH suele ser la opción por defecto, esto se debe a su simplicidad y seguridad.

Sin embargo, existen otras herramientas que nos pueden ayudar a llevar a cabo la misma tarea, por ejemplo, si buscamos tener un entorno gráfico, existen varias alternativas que pueden complementar o reemplazar a SSH.

Entre ellas encontramos X11 Forwarding, RDP en Linux, y VNC.

X11 Forwarding, permite ejecutar aplicaciones gráficas de forma remota en una máquina y mostrarlas localmente a través del protocolo X11, emplea una conexión SSH como canal seguro, lo que nos garantiza la seguridad además de poder ejecutar aplicaciones gráficas remotamente. Suele recomendarse para tareas puntuales y no para sesiones completas de escritorio remoto.

RDP es nativo de Windows, aunque existen servidores compatibles con RDP en Linux como xrdp, que permiten conexiones desde clientes RDP para acceder a un escritorio completo, esta opción será explorada y detallada en la segunda sección del trabajo utilizando Remmina.

VNC es un protocolo de escritorio remoto que transmite la pantalla y permite controlar otro equipo gráficamente. Permite el acceso completo al escritorio gráfico remoto, aunque no cifra las conexiones de forma predeterminada.

En conclusión, según nuestras necesidades podemos utilizar un método u otro, ya que no parece que ninguno tenga una ventaja clara sobre el resto. De esta forma, parece que la mejor solución sería combinar algunas de estas herramientas con SSH para obtener túneles cifrados y así proteger la conexión.

3. Acceso Remoto Linux a Windows

En esta segunda parte del trabajo realizaremos una conexión entre una máquina Linux y una Windows utilizando RDP, que es el sistema que usa Windows por defecto para permitir conexiones remotas a su escritorio.

Desde Linux, una de las herramientas más prácticas para conectarse a través de RDP es Remmina. Remmina es una aplicación de escritorio remoto potente y fácil de usar, que funciona con varios protocolos como RDP, VNC, SSH, entre otros. Nos permite crear conexiones, guardarlas, ajustar la calidad de imagen, la resolución e incluso el comportamiento del ratón y del teclado.

RDP se encarga de transmitir la imagen del escritorio de la máquina Windows y recibe las acciones que hacemos desde Linux. Esto nos permite manejar la otra máquina como si estuviéramos en ella.

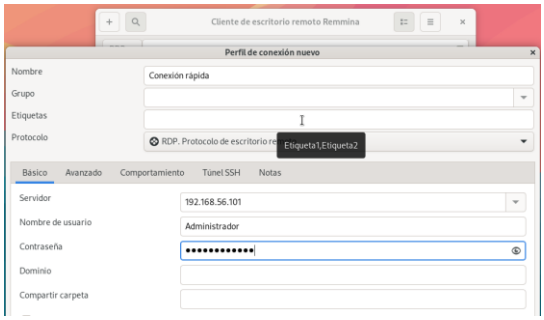
A continuación, se mostrará como establecer una conexión entre Linux y Windows utilizando Remmina y RDP.

3.1 RDP utilizando Remmina

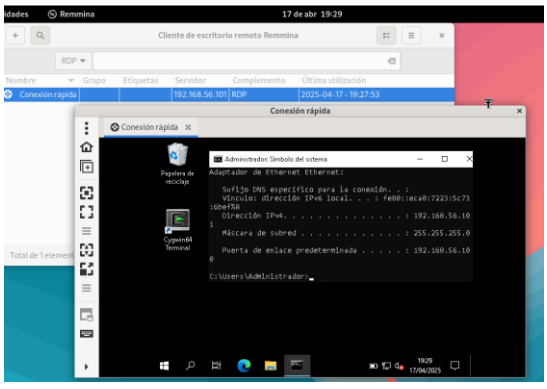
El primer paso es instalar un cliente RDP en AlmaLinux, como ya hemos mencionado anteriormente, en este ejemplo se utilizará Remmina. Ejecutamos “dnf install remmina remmina-plugins-rdp"

Después, tendremos que configurar Windows Server 22 para que se permita el acceso remoto:

Tras haber configurado Windows Server Correctamente, abrimos Remmina en nuestra máquina Linux y tratamos de establecer la conexión:



Finalmente, después de haber realizado la configuración inicial, nos conectamos:



3.2 Alternativas

Otra opción que podríamos utilizar para establecer una conexión remota entre una máquina Linux y otra Windows sería WinRM.

Tanto Remmina con RDP como WinRM permiten gestionar y controlar sistemas Windows de forma remota, pero cada uno tiene características distintas que los hacen más adecuados según el caso de uso.

WinRM es una tecnología de Microsoft basada en el estándar WS-Management, que permite ejecutar comandos y scripts remotamente. En Linux, se puede usar mediante herramientas como pywinrm o Ansible. A diferencia de Remmina, WinRM no nos proporciona interfaz gráfica.

En conclusión, ambas opciones son válidas, pero están pensadas para distintos usos. RDP con Remmina es ideal para tareas que requieran GUI, a diferencia de WinRM, que es excelente para administración remota sin necesidad de interfaz gráfica.

4. Acceso Remoto de Windows a Windows

En esta sección del trabajo utilizaremos PowerShell Remoting, esta tecnología permite ejecutar comandos y scripts de PowerShell en otras máquinas dentro de la red, sin necesidad de acceder físicamente a ellas ni abrir sesiones gráficas.

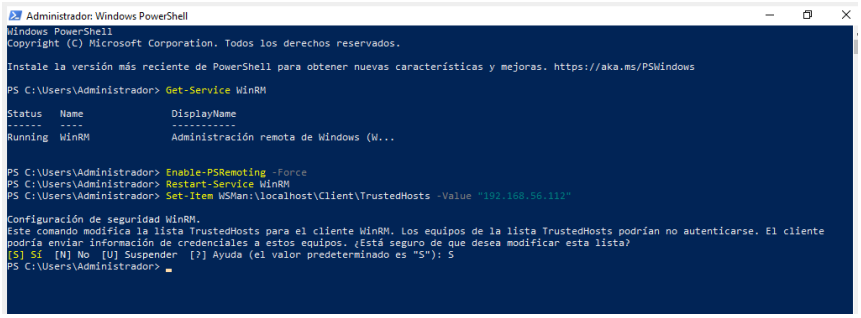
PowerShell Remoting se basa en el protocolo WS-Management, y permite establecer sesiones remotas seguras y encriptadas entre equipos. Está integrado de forma nativa en Windows.

En esta sección se explorará cómo establecer una conexión remota entre dos equipos Windows utilizando PowerShell Remoting.

4.1 Acceso Remoto con PowerShell Remoting

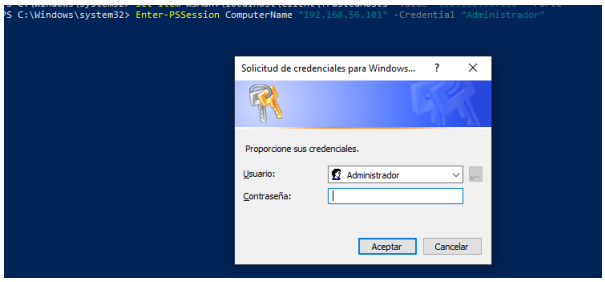
Usaremos 2 máquinas virtuales, Windows server 2022 y Windows 10.

Primero, configuramos Windows Server 2022:

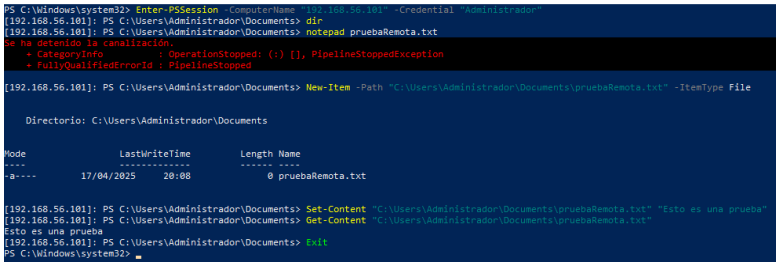


Tras realizar la configuración del servidor configuramos nuestra máquina Windows 10 para poder realizar la conexión de forma correcta.

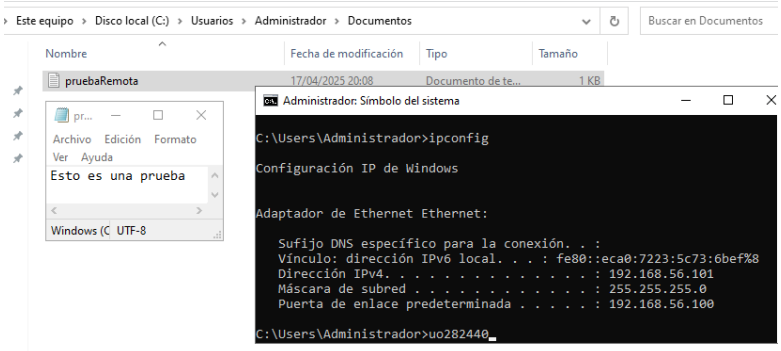
Nos conectamos a la maquina Windows server 2022:



Creamos un archivo de forma remota para posteriormente comprobar que se muestra en el servidor:



Comprobamos que se muestra adecuadamente:



4.2 Alternativas

Aunque PowerShell Remoting es una herramienta potente para la administración remota entre sistemas Windows, existen otras alternativas más orientadas al control visual del equipo, como es el caso de TeamViewer. Ambas opciones permiten conectarse remotamente a otro equipo, pero están diseñadas con propósitos diferentes y ofrecen experiencias muy distintas.

TeamViewer permite ver y controlar el escritorio remoto de forma completa, con una interfaz gráfica. Es multiplataforma y accesible desde Windows, Linux, macOS e incluso móviles.

Aunque, algo a tener en cuenta, es que es un software de terceros, y su versión gratuita tiene limitaciones de uso.

Si el objetivo es administrar equipos Windows de forma más profesional, automatizar tareas o ejecutar scripts de forma remota en una red local o de empresa, probablemente PowerShell Remoting sea mejor opción.

Si, por el contrario, se necesita revisar visualmente el escritorio o dar asistencia remota a otra persona, incluso a través de internet, Team Viewer podría ser una mejor opción.

5. Conclusión

Este trabajo me ha permitido explorar y aprender sobre diferentes formas de realizar conexiones remotas en diferentes contextos, en función de las necesidades del momento, y sin importar los sistemas operativos involucrados, ya que se han visto diferentes alternativas para realizar conexiones remotas tanto en Linux como en Windows.