

Actividad Práctica: Implementación de un Servidor SSH Seguro en Windows

Introducción y Escenario

Hoy asumes el rol de especialista en seguridad de sistemas en la empresa "SeguTech". Tu compañero Benito, un desarrollador clave, acaba de recibir un nuevo servidor Windows que necesita para su próximo proyecto. Para que pueda trabajar de forma remota y segura, te han encargado una tarea crucial: **habilitar y configurar el acceso remoto seguro a su nuevo equipo.**

Por supuesto, protocolos antiguos como Telnet están totalmente prohibidos por la política de seguridad de la empresa debido a sus graves vulnerabilidades. Tu misión es implementar la solución estándar de la industria: el protocolo **SSH (Secure Shell)**.

A través de esta guía, convertirás el equipo de Benito en un servidor SSH, lo configurarás correctamente y realizarás la primera conexión segura para verificar que todo funciona a la perfección. ¡Manos a la obra!

Relación con el Módulo MF0489_3

Esta actividad te permitirá aplicar de forma práctica los conceptos vistos en el manual:

- **Protocolos SSL/TLS y SSH:** Implementarás uno de los protocolos de comunicación segura más importantes del mundo.
- **Túneles Cifrados:** SSH crea un túnel cifrado para proteger la sesión de administración remota, garantizando la confidencialidad e integridad de los comandos y sus respuestas.

Objetivos de Aprendizaje

- Instalar y habilitar el componente "Servidor OpenSSH" en un sistema Windows.
- Configurar las reglas del Firewall de Windows para permitir conexiones SSH entrantes.
- Utilizar un cliente SSH para establecer una conexión remota segura.
- Comprender la importancia de la autenticidad del servidor a través de su huella digital.

Tarea a Realizar

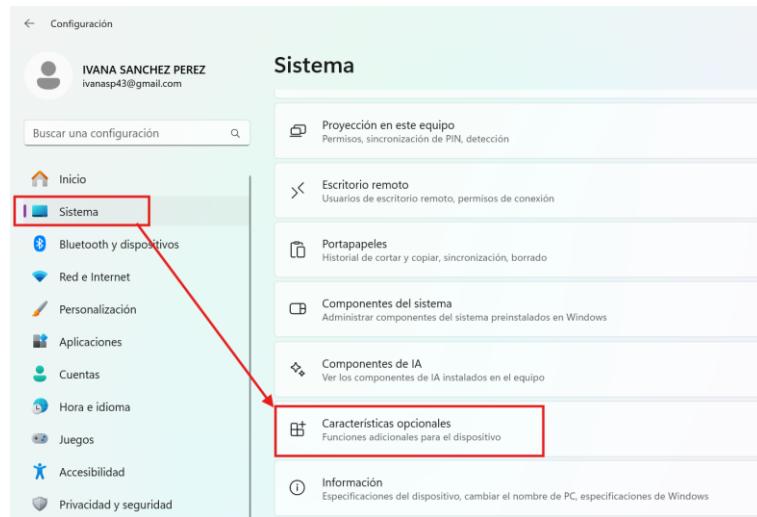
Sigue los pasos detallados en tu manual para configurar el servidor SSH y responde a las preguntas de reflexión a medida que avanzas.

1. Instalación del Servidor OpenSSH (Consulta las páginas 146-150 del manual en PDF, capítulo 3, tema 2.1)

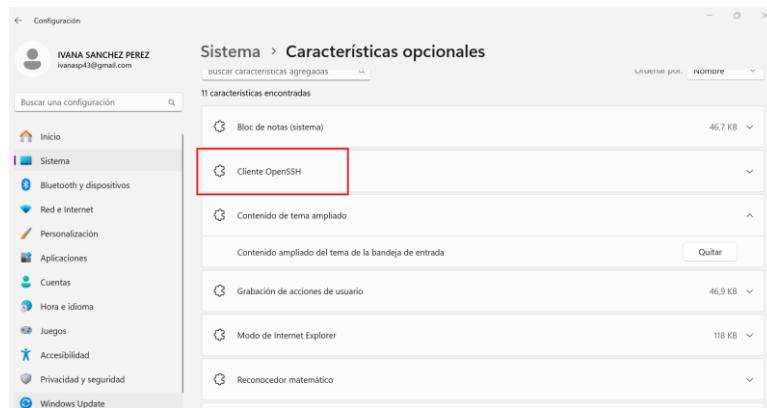
- Sigue los pasos descritos para instalar la característica opcional "**Servidor OpenSSH**" en el sistema Windows.

MF0489_3 Comunicaciones Seguras

- 1- Abrir la configuración de Windows → Inicio y seleccionar Configuración
- 2- Acceder a las características opcionales → Sistema > Características opcionales

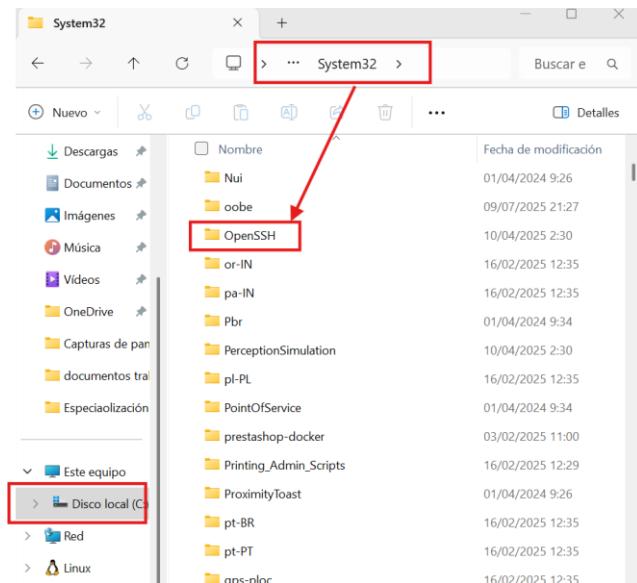


- 3- Ver las características instaladas



- 4- Activar el cliente OpneSSH
- 5- Agregar una característica opcional
- 6- Buscar SSH en las características
- 7- Activar el servidor Open SSH
- 8- Instalar el servidor SSH
- 9- Finalizar la instalación
- 10- Verificación de la instalación → C:\Windows\system32\Openssh

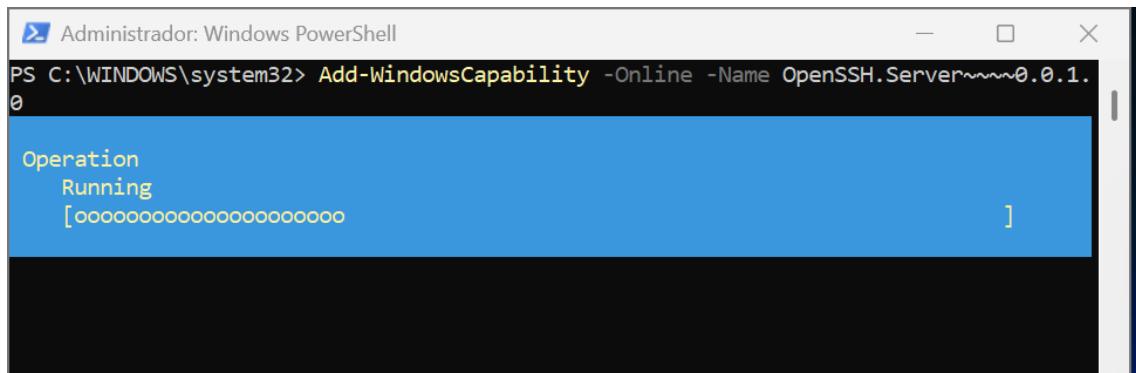
MF0489_3 Comunicaciones Seguras



Otra forma más rápida es a través del terminal de la PowerShell

- 1- Abre PowerShell como **Administrador**
- 2- Ejecutamos el siguiente comando para instalar el servidor OpenSSH

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```



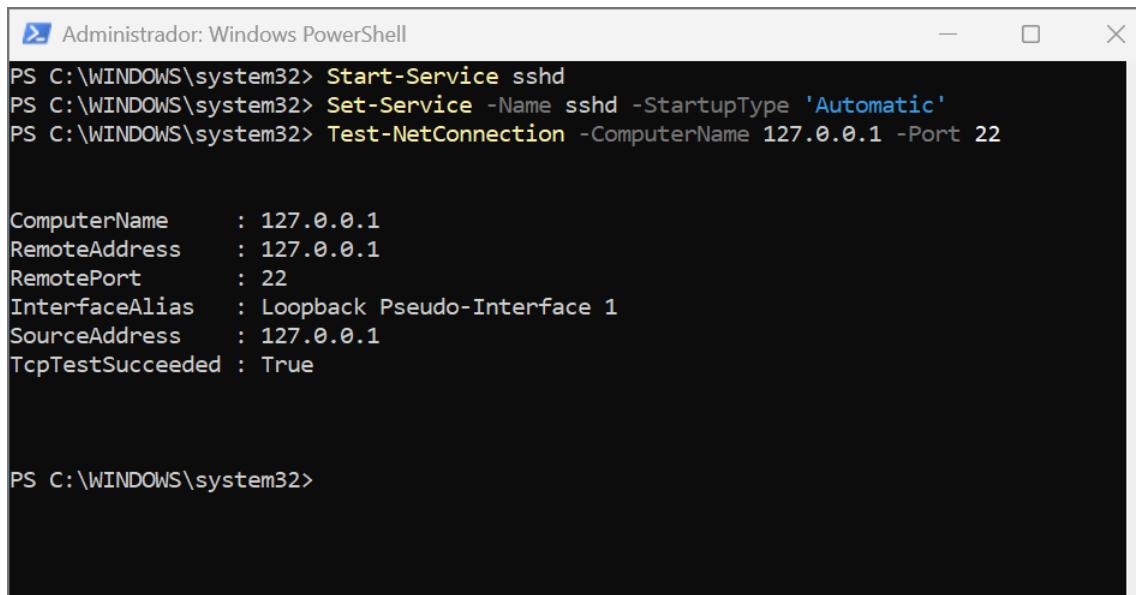
- 3- Esperar que la instalación complete
- 4- Una vez que se haya instalado, iniciamos el servicio y lo configuramos para que se inicie automáticamente.

```
Start-Service sshd  
Set-Service -Name sshd -StartupType 'Automatic'
```

- 5- Ejecutamos la prueba de conexión. Ya no hace falta la regla de firewall, pues ya el instalador de Open SSH lo hace automáticamente.

```
Test-NetConnection -ComputerName 127.0.0.1 -Port 22
```

```
Start-Service sshd  
Set-Service -Name sshd -StartupType 'Automatic'
```



The screenshot shows a Windows PowerShell window titled "Administrador: Windows PowerShell". The command PS C:\WINDOWS\system32> Start-Service sshd is run, followed by PS C:\WINDOWS\system32> Set-Service -Name sshd -StartupType 'Automatic'. Then, PS C:\WINDOWS\system32> Test-NetConnection -ComputerName 127.0.0.1 -Port 22 is run, which outputs the following results:

```
ComputerName      : 127.0.0.1
RemoteAddress    : 127.0.0.1
RemotePort       : 22
InterfaceAlias   : Loopback Pseudo-Interface 1
SourceAddress    : 127.0.0.1
TcpTestSucceeded : True
```

Finally, PS C:\WINDOWS\system32> is shown at the bottom.

- Pregunta de Reflexión 1:

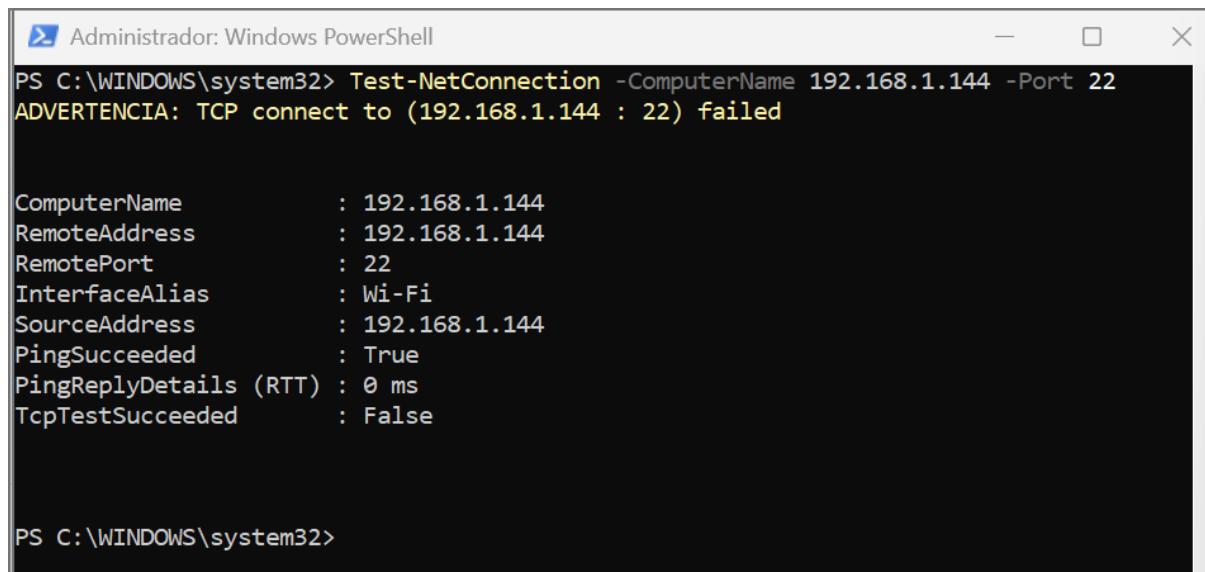
Estás implementando SSH. ¿Por qué este protocolo es una alternativa mucho más segura que protocolos antiguos como Telnet para la administración remota? ¿Qué propiedad de seguridad clave ofrece SSH de la que Telnet carece?

SSH es mucho más seguro que Telnet porque **establece un canal cifrado** para la comunicación, protegiendo la confidencialidad y la integridad de los datos que se transmiten. Telnet envía la información, incluidas las contraseñas, en texto plano, lo que puede ser interceptado fácilmente por atacantes. La propiedad clave que ofrece SSH y que Telnet no tiene es el **cifrado fuerte de toda la sesión de administración remota**.

2. Configuración del Firewall (Consulta la página 171 del manual)

- Asegúrate de que el firewall de Windows permite las conexiones entrantes en el **puerto TCP 22**, que es el estándar para SSH.

MF0489_3 Comunicaciones Seguras

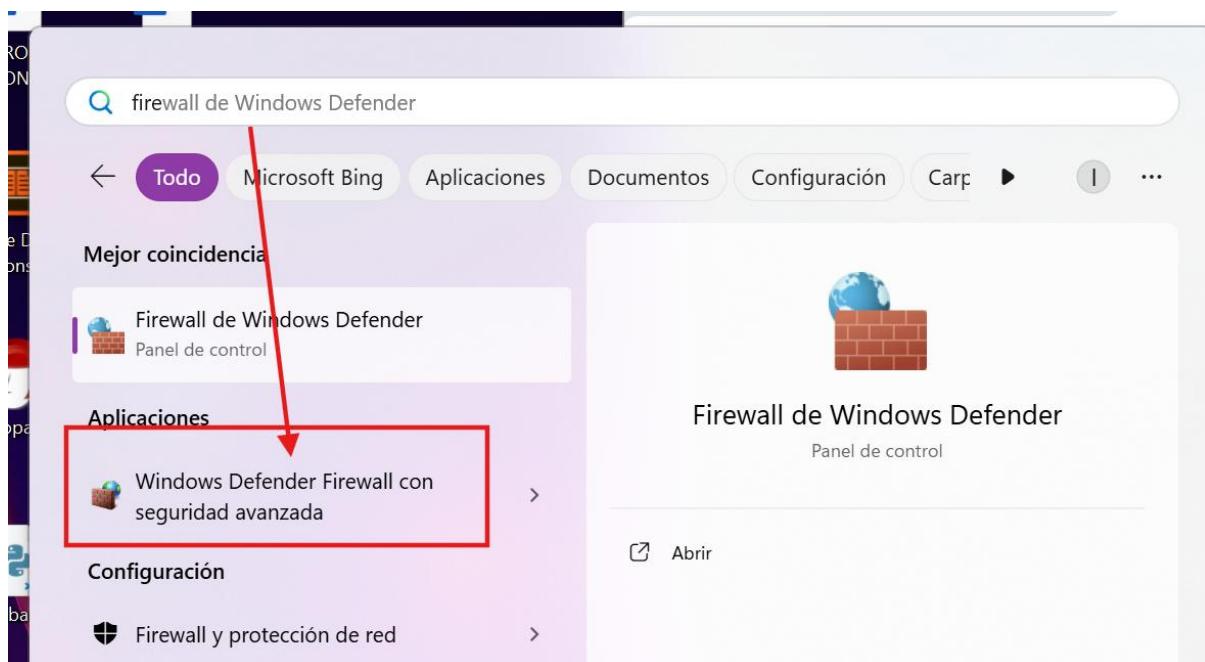


```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Test-NetConnection -ComputerName 192.168.1.144 -Port 22
ADVERTENCIA: TCP connect to (192.168.1.144 : 22) failed

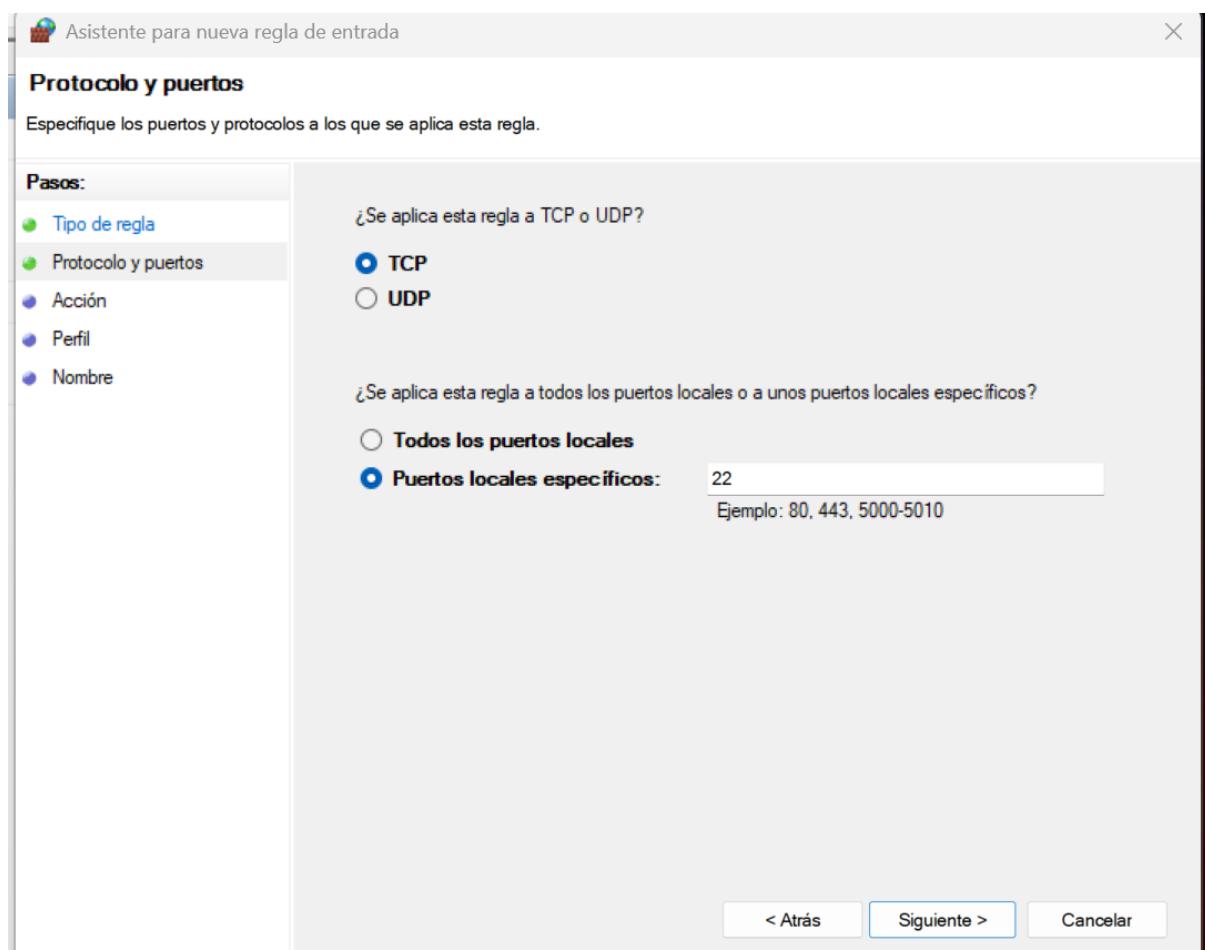
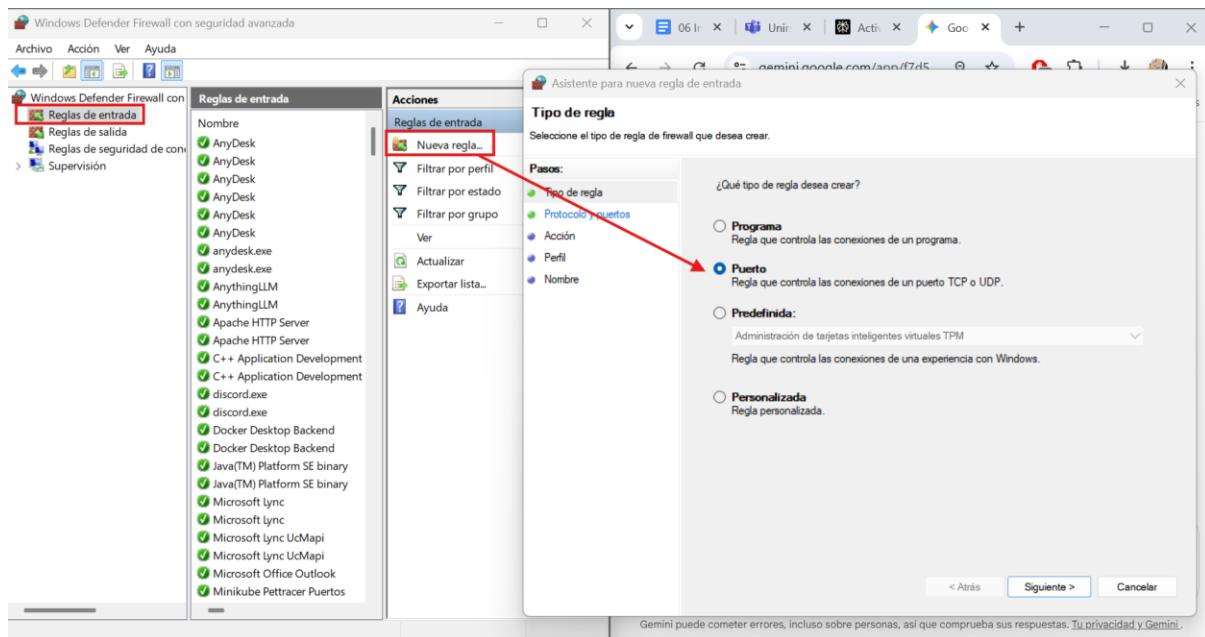
ComputerName      : 192.168.1.144
RemoteAddress     : 192.168.1.144
RemotePort        : 22
InterfaceAlias    : Wi-Fi
SourceAddress     : 192.168.1.144
PingSucceeded     : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded   : False

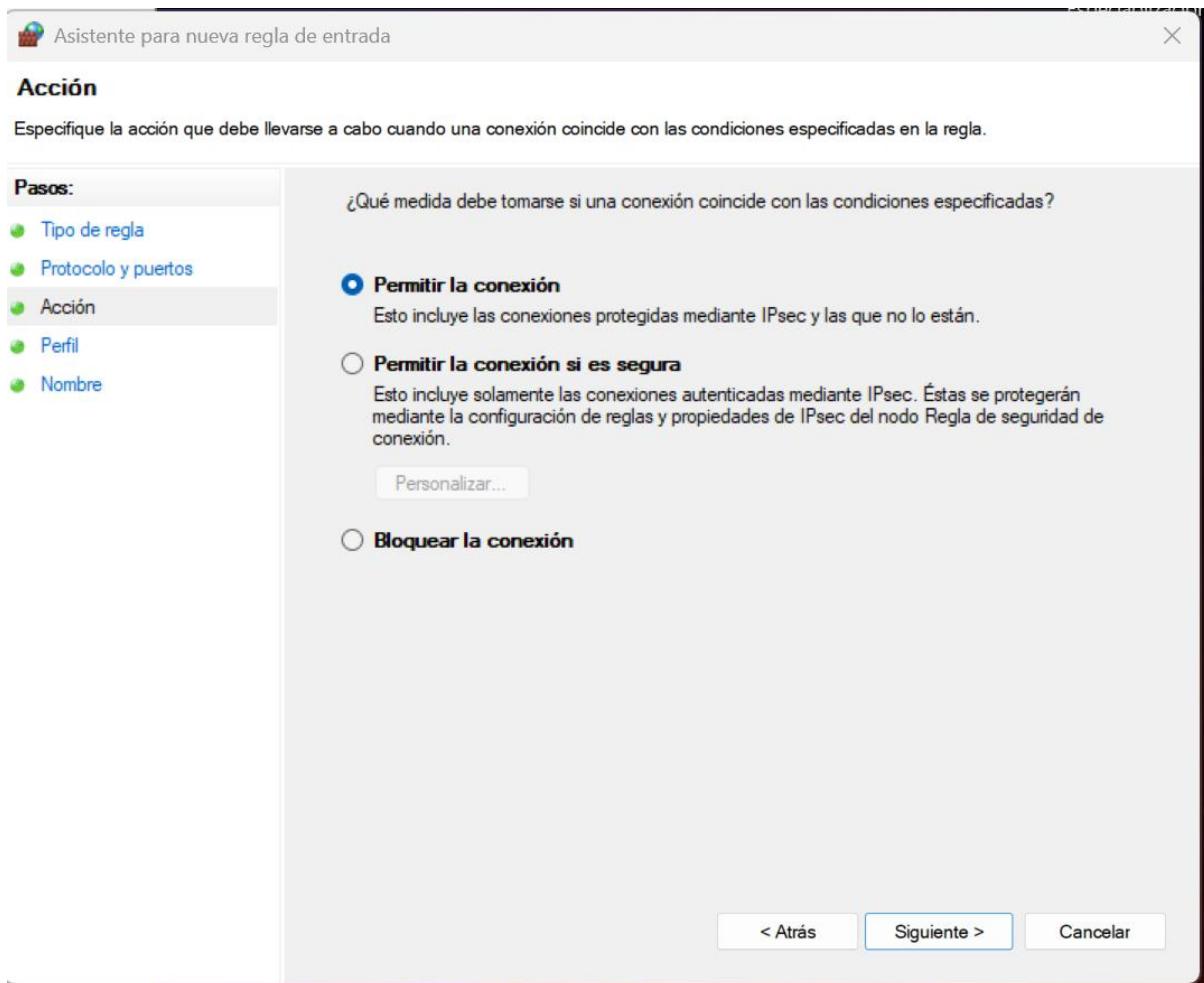
PS C:\WINDOWS\system32>
```

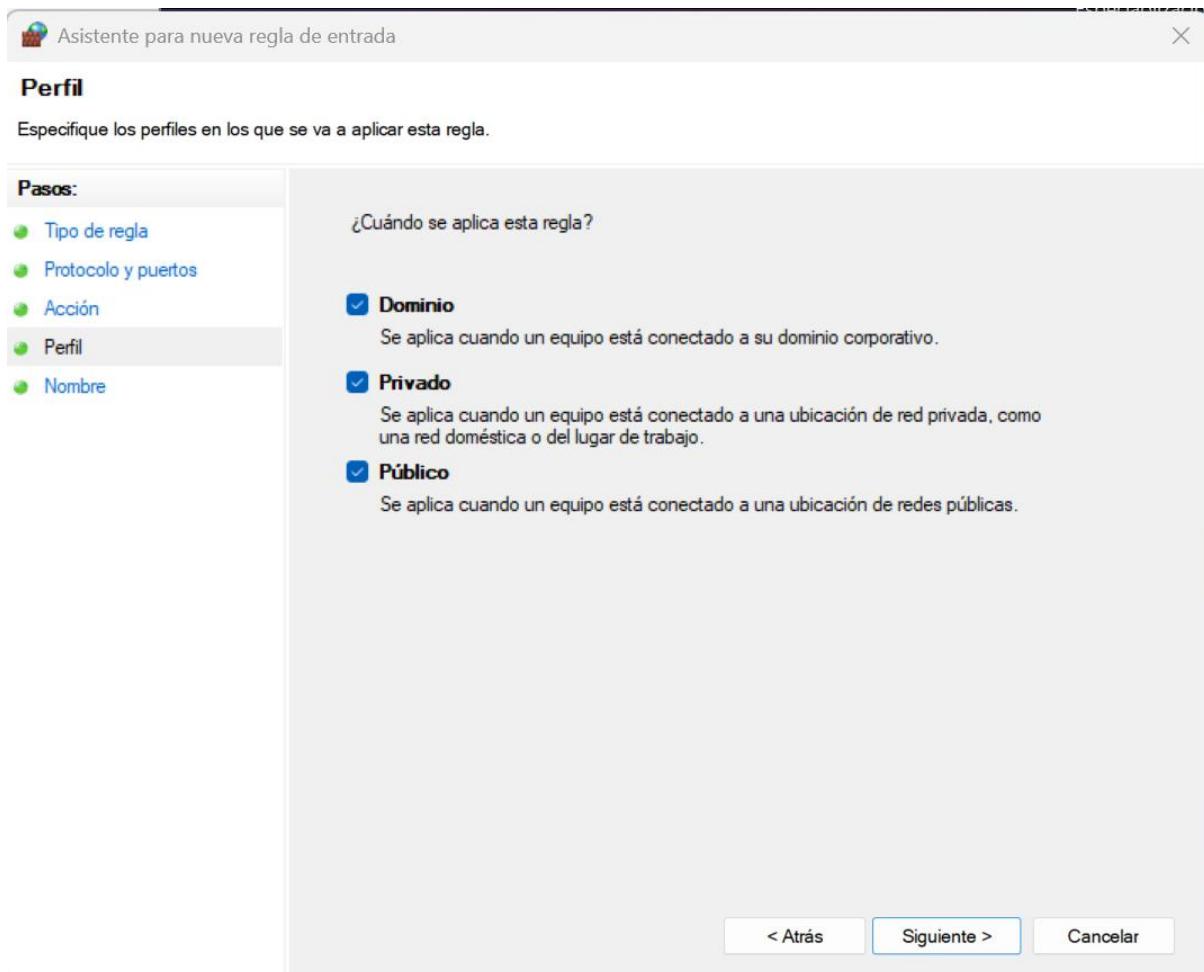
El puerto 22 ahora mismo no está abierto



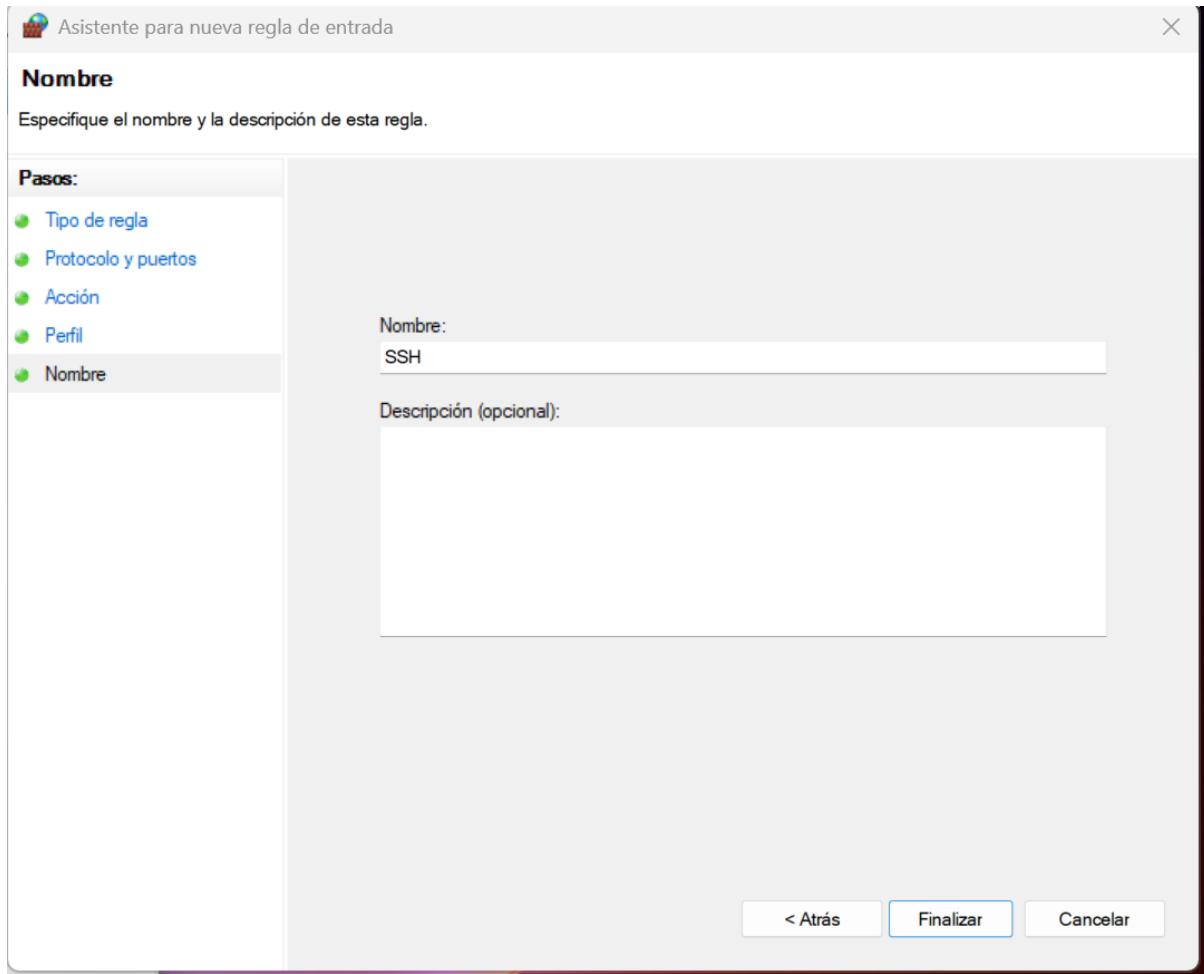
MF0489_3 Comunicaciones Seguras





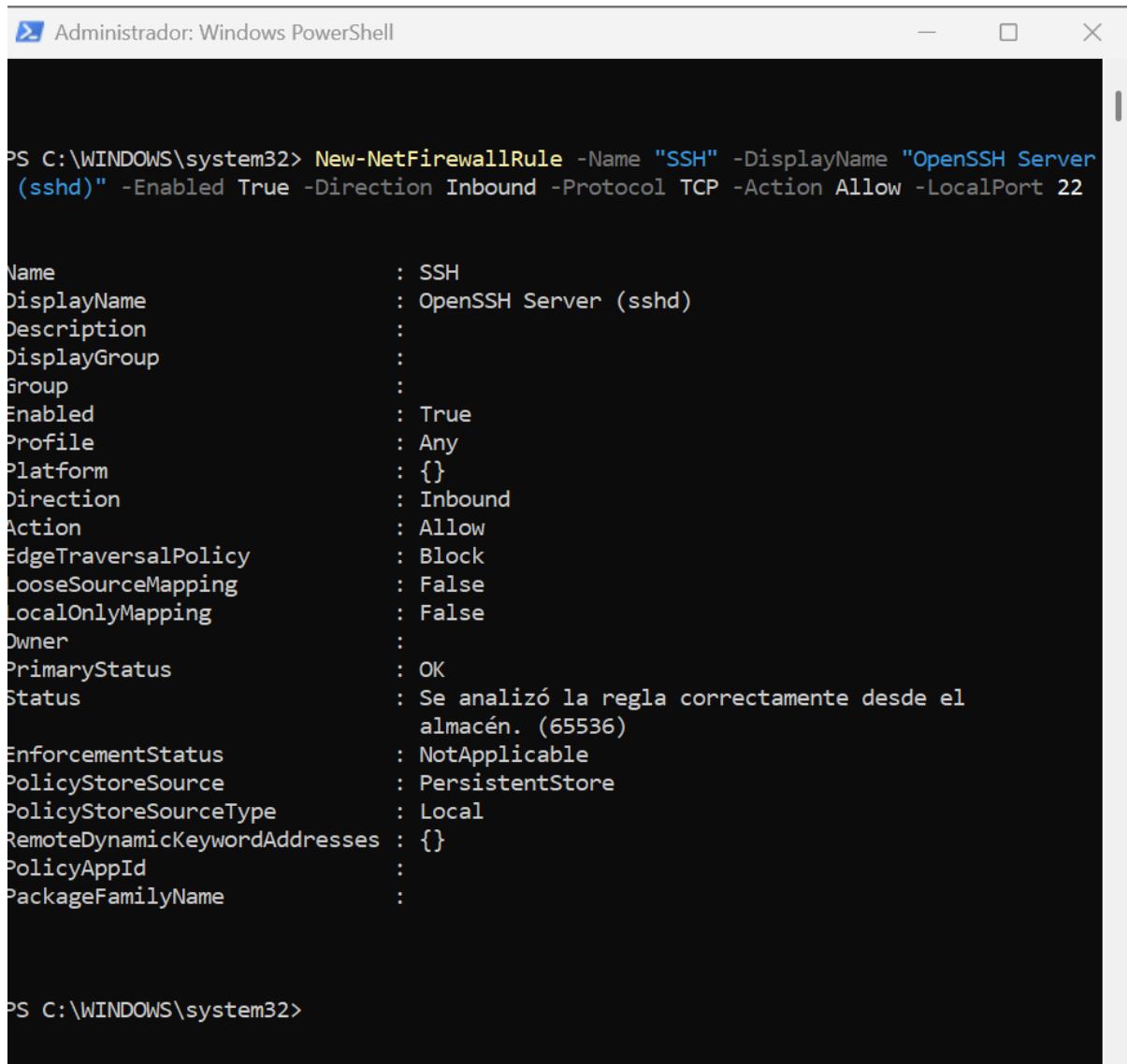


MF0489_3 Comunicaciones Seguras



Creo la regla de firewall para abrir el puerto

MF0489_3 Comunicaciones Seguras



The screenshot shows a Windows PowerShell window titled "Administrador: Windows PowerShell". The command executed is:

```
PS C:\WINDOWS\system32> New-NetFirewallRule -Name "SSH" -DisplayName "OpenSSH Server (sshd)" -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
```

The output displays the properties of the newly created firewall rule:

Name	:	SSH
DisplayName	:	OpenSSH Server (sshd)
Description	:	
DisplayGroup	:	
Group	:	
Enabled	:	True
Profile	:	Any
Platform	:	{}
Direction	:	Inbound
Action	:	Allow
EdgeTraversalPolicy	:	Block
LooseSourceMapping	:	False
LocalOnlyMapping	:	False
Owner	:	
PrimaryStatus	:	OK
Status	:	Se analizó la regla correctamente desde el almacén. (65536)
EnforcementStatus	:	NotApplicable
PolicyStoreSource	:	PersistentStore
PolicyStoreSourceType	:	Local
RemoteDynamicKeywordAddresses	:	{}
PolicyAppId	:	
PackageFamilyName	:	

At the bottom of the window, the prompt "PS C:\WINDOWS\system32>" is visible.

MF0489_3 Comunicaciones Seguras

```
Administrator: Windows PowerShell
mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction SilentlyContinue)) {
>>     New-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -DisplayName "OpenSSH Server (sshd)" -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
>> } else {
>>     Write-Output "La regla de firewall para SSH ya existe."
>> }

Name          : OpenSSH-Server-In-TCP
DisplayName   : OpenSSH Server (sshd)
Description   :
DisplayGroup :
Group        :
Enabled       : True
Profile       : Any
Platform      : {}
Direction    : Inbound
Action        : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner         :
PrimaryStatus : OK
Status        : Se analizó la regla correctamente desde el
                almacén. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId   :
PackageFamilyName :
```

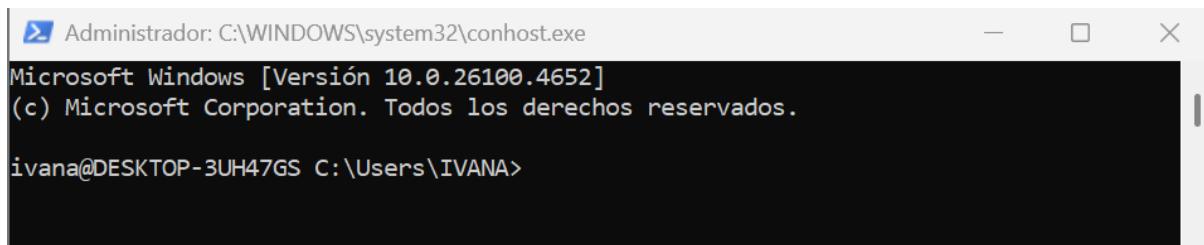
Con el comando **netstat** se mostrarán todas las conexiones y puertos que usan el número 22.

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> netstat -an | findstr :22
TCP    0.0.0.0:22          0.0.0.0:0              LISTENING
TCP    [::]:22             [::]:0                LISTENING
PS C:\WINDOWS\system32>
```

Estas líneas significan que el puerto 22 está abierto y el servidor está escuchando conexiones-

3. Prueba de Conexión Local

- Abre una terminal de Windows (PowerShell o CMD).
- Establece una conexión SSH contra tu propio equipo usando el comando `ssh tu_nombre_de_usuario@localhost`.
- La primera vez que te conectes, el cliente te mostrará la "huella digital" (fingerprint) de la clave del servidor y te preguntará si confías en él. Escribe `yes` para continuar.



```
Administrator: C:\WINDOWS\system32\conhost.exe
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

ivana@DESKTOP-3UH47GS C:\Users\IVANA>
```

Pregunta de Reflexión 2:

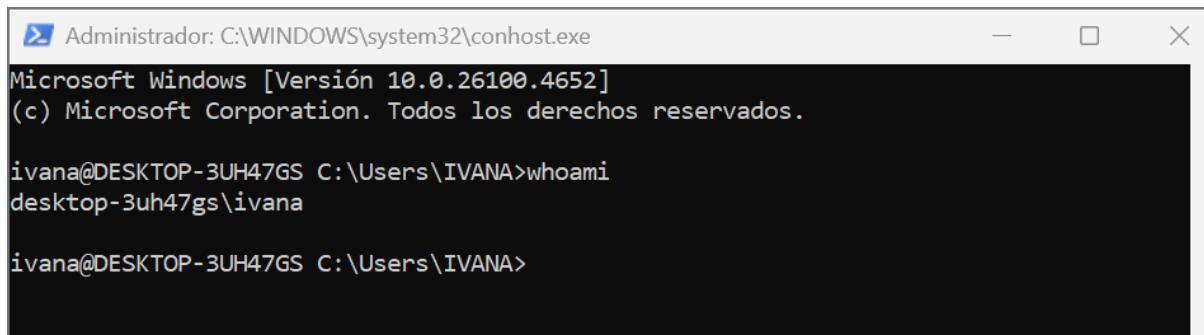
El terminal te ha mostrado una "huella digital" (fingerprint) y te ha preguntado si quieras continuar. ¿Qué representa esta huella digital y por qué es un paso de seguridad tan importante la primera vez que te conectas a un servidor SSH nuevo?

La "huella digital" es el **hash único de la clave pública del servidor SSH**. Representa la identidad del servidor. Confirmar esta huella la primera vez garantiza que no estás siendo víctima de un ataque de hombre en medio (Man-in-the-Middle), donde alguien se haría pasar por el servidor. Es un paso crítico para verificar la autenticidad y garantizar que la conexión se establece con el servidor correcto.

- Introduce tu contraseña de inicio de sesión de Windows para autenticarte.

4. Verificación y Documentación

- Una vez dentro de la sesión remota, ejecuta un comando simple como `whoami` o `dir` para demostrar que la conexión es funcional.



```
Administrator: C:\WINDOWS\system32\conhost.exe
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

ivana@DESKTOP-3UH47GS C:\Users\IVANA>whoami
desktop-3uh47gs\ivana

ivana@DESKTOP-3UH47GS C:\Users\IVANA>
```

Pregunta de Reflexión 3:

Has abierto el puerto 22 en el firewall para permitir el acceso. Si este servidor estuviera directamente conectado a Internet, ¿qué riesgo principal supondría tener el puerto 22 abierto para todo el mundo? ¿Qué medida de seguridad adicional (que no hemos implementado en esta práctica básica) sería crucial para proteger el acceso?

Si el puerto 22 está abierto al mundo en un servidor directamente conectado a Internet, el riesgo principal es que **atacantes pueden intentar ataques de fuerza bruta u otros tipos de intrusión para acceder al sistema.**

Una medida crucial para proteger el servidor sería implementar autenticación mediante **llaves SSH** en lugar de solo contraseña, además de usar listas blancas de IP o VPNs, y otras medidas como cambiar el puerto por defecto o usar herramientas de bloqueo tras varios intentos fallidos.

Entregable

- **Formato:** Un (1) único fichero en formato **PDF**.
- **Nombre del Fichero:** MF0489_Actividad_SSH_NombreApellido.pdf
- **Contenido del Informe:**
 1. **Portada:** Con tus datos y el nombre de la actividad.
 2. **Verificación de Instalación:** Una captura de pantalla de la sección "Características opcionales" de Windows donde se vea que el "Servidor OpenSSH" está instalado.
 3. **Configuración del Firewall:** Una captura de pantalla de la configuración del Firewall de Windows Defender donde se aprecie la regla que permite el tráfico entrante para el puerto 22.
 4. **Prueba de Conexión:** Una captura de pantalla completa de tu terminal mostrando el comando de conexión a `localhost`, la pregunta sobre la huella digital y la salida del comando `whoami`.
 5. **Respuestas a las Preguntas de Reflexión:** Una sección donde respondas de forma clara y razonada a las tres preguntas planteadas durante la actividad.