SSH

Introducción
OpenSSH
Túneles
Autenticación

SSH

SSH es un protocolo pero también puede ser visto o analizado como una herramienta para encriptar, una aplicación cliente/servidor o una interfaz de comandos.

En su forma más habitual o común SSH puede ser visto como un reemplazo seguro para Telnet, ya que tanto los comandos como los datos viajan encriptados.
Aunque SSH es mucho más potente y versátil que un mero login remoto encriptado.

SSH

El protocolo SSH ofrece: autenticación , encriptación e integridad

"soy mmoss@pampero.itba.edu.ar"

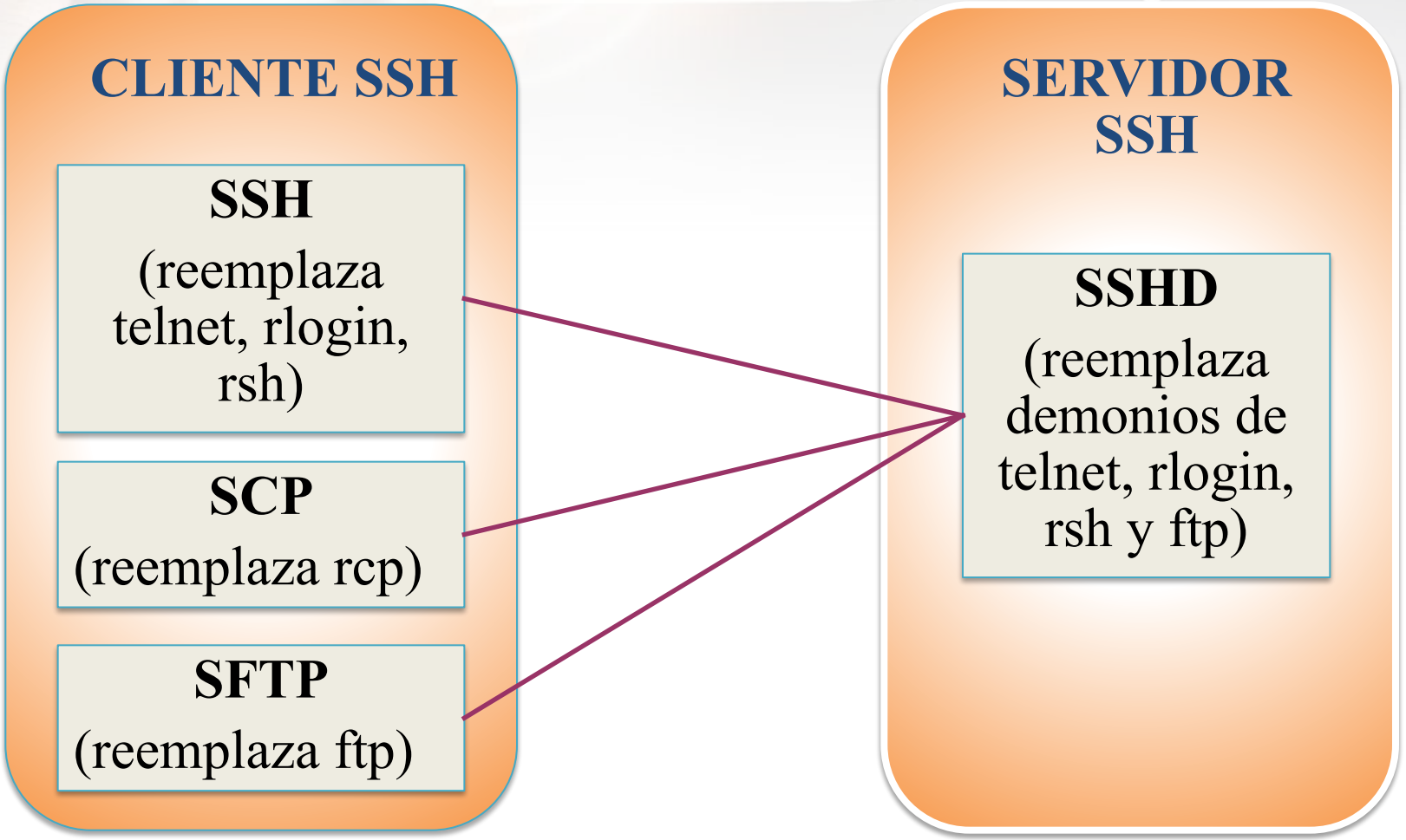
"y yo soy pampero.itba.edu.ar"

"hola mundo" → "#@%6h/&%=?¡" → "hola mundo"

"hola mundo" → "#@%6h/XXX?¡" **¡Mensaje alterado!**

SSH

SSH es el reemplazo seguro de varias aplicaciones



SSH

Los principales servidores SSH son:

- OpenSSH: Windows, Unix y Linux
- SSH Communications: Windows, Unix y Linux
- VanDyke's Vsheel, BitVise: Windows.

Nosotros utilizaremos OpenSSH, que viene instalado por defecto en las distribuciones Linux.
El demonio que utiliza es **sshd** y escucha, por defecto, en el puerto 22.

OpenSSH

Componentes básicos de OpenSSH

- sshd: Es el demonio que maneja las conexiones entrantes
  - No debe ser iniciado por inetd o xinetd
- ssh: Comando que reemplaza a otros como rsh y rlogin,
- para loguearse en forma remota o enviar un comando
- a un servidor remoto.
- scp: Versión segura de rcp, que permite copiar archivos
- sftp: Versión segura de FTP
- ssh-keygen: generador de claves públicas y privadas.

OpenSSH

El archivo principal de configuración es **`/etc/sshd_config`**.
Algunas de las opciones generales son:

```
#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::

#LoginGraceTime 120
PermitRootLogin no
#StrictModes yes
```

Ejemplos de uso

Sesión de terminal remota

```
user@server:~$ ssh pi@pampero.itba.edu.ar
Password: ******
Last Login Wed May 4 16:12:22 from ....
Linux 2.4.25-lids1.2.0rc2
user@server:~$ ...
user@server:~$ logout
```

La misma interacción que con telnet

Ejecución remota de un comando

```
user@server:~$ ssh user@host rm /tmp -r
```

Ejemplos de uso

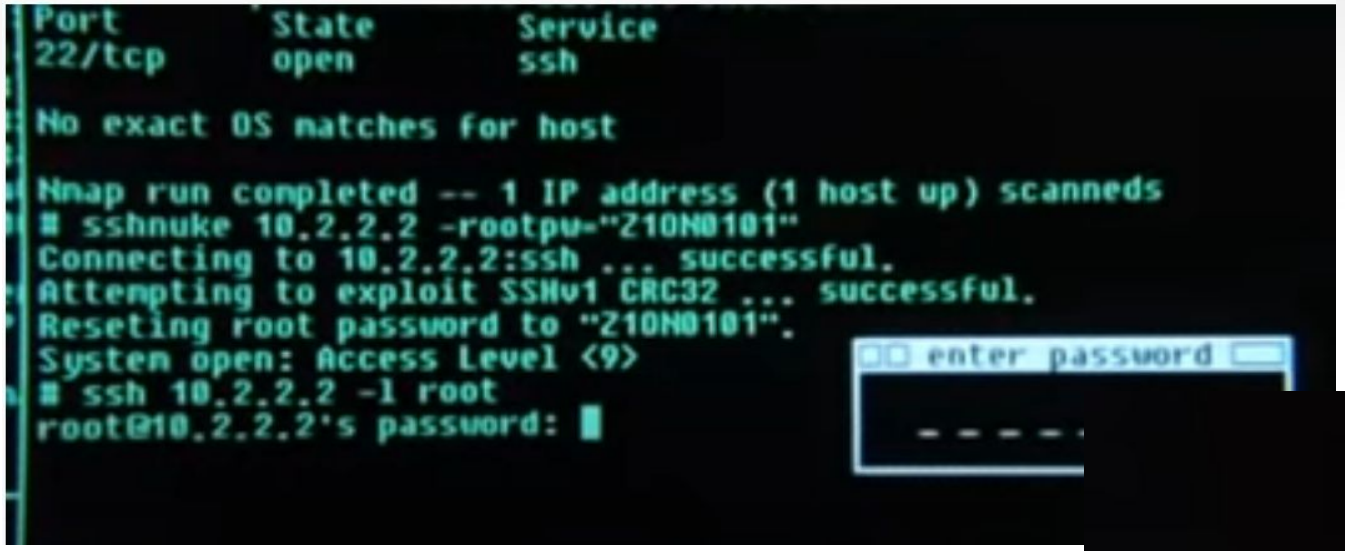
Sesión de terminal remota

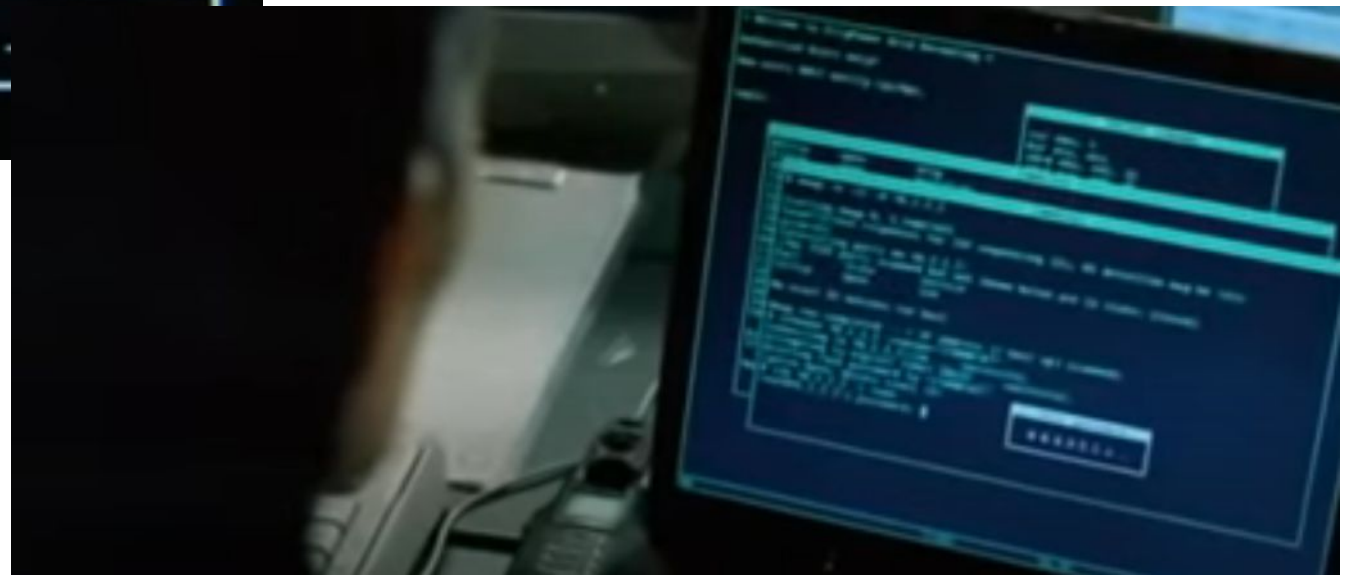
Algunas de las opciones para ssh son:

- -1 Fuerza ssh a usar SSH versión 1.
- -2 Fuerza ssh a usar SSH versión 2.
- -C Comprime los datos
- -c blowfish | 3des | desElegir método de encriptación
- -v Muestra información de depuración
- -4 Forzar a IP v4
- -6 Usar únicamente IP v6
- -E *log_file* Info de debug a un archivo
- -p *port* Usar un puerto distinto a 22
- -N Sólo conecta, no ejecuta comando remoto

SSH Versión 1

Las primeras versiones de OpenSSH 1 contenían errores de implementación que la hacían vulnerable, como bien lo sabía Trinity





Ejemplos de uso

Copiar archivos

```
mgarbe@pampero:~$ scp yo@casa.no-ip.info:*.pdf .
Password: ******
parcial1.pdf100% 9428 9.2KB/s   00:00
parcial2.pdf100% 8508 1.5KB/s   00:01
final.pdf    100% 7099 1.1KB/s   00:00
```

Además de las opciones vistas para ssh, scp incluye:

- -q Deshabilita porcentaje de progreso
- -r Copia recursivamente un directorio

Ejemplos de uso

FTP seguro

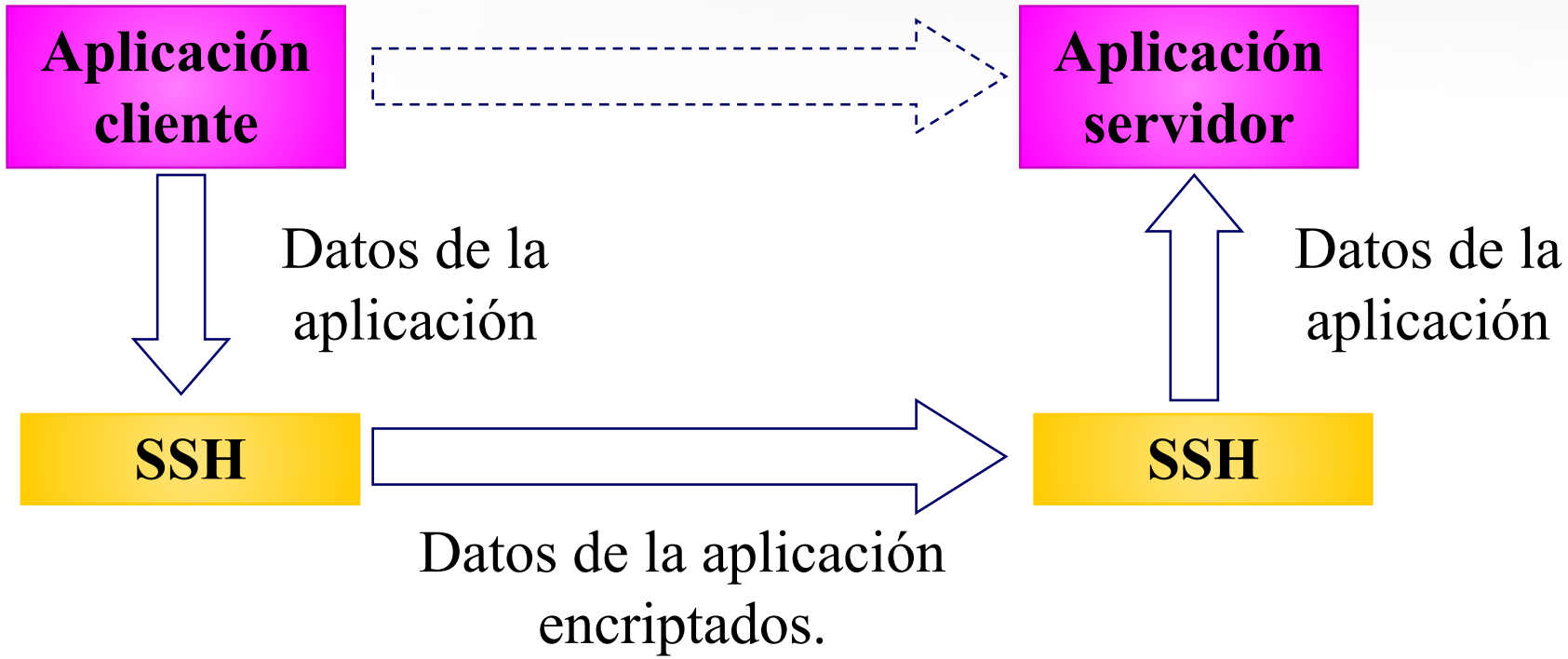
```
user@server:~$ sftp mgarbe@pampero.itba.edu.ar
Password: ******
sftp> lls
sftp> lpwd
sftp> quit
```

Alguna opción particular de sftp:

- -B *buffer_size* Buffer de transferencia.
- -b *batch_file* Lee los comandos desde un archivo y no desde stdin.

Port forwarding

SSH puede encriptar en forma transparente el flujo de datos de otra aplicación (*tunneling*)



Port forwarding

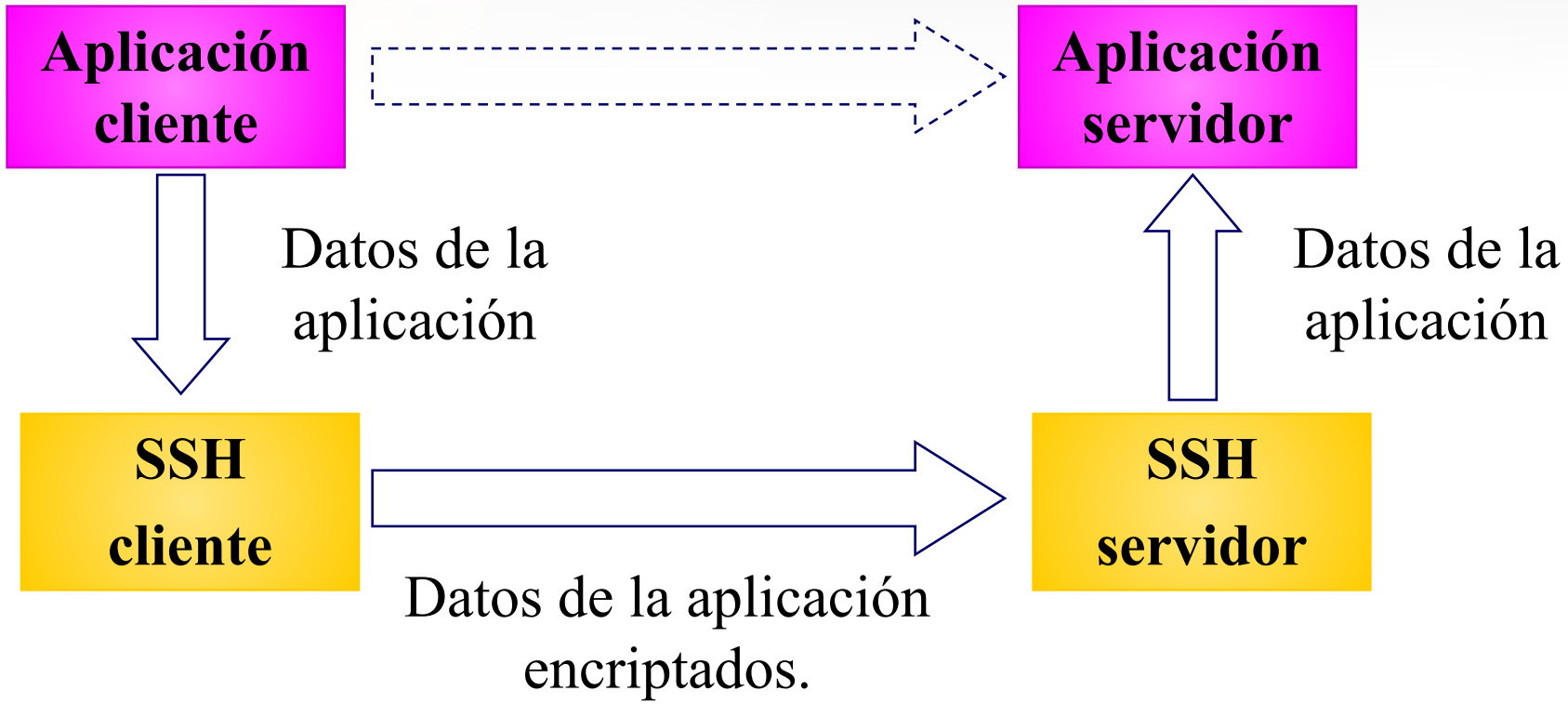
SSH port forwarding protege conexiones TCP redirigiéndolas a través de una conexión SSH. Una vez establecida la conexión se le pueden dar diversos usos, como por ejemplo:

- Acceder a servicios TCP (SMTP, IMAP, POP, LDAP, etc.) a través de un firewall que no permita conexiones a ellos.
- Protege la sesión con estos servicios, encriptando claves y datos.
- Enviar mensajes usando un servidor SMTP que no permita relying, trabajando en otra red.

Notar que funciona sólo con TCP, no funciona con protocolos basados en UDP como DNS, DHCP, NFS, NetBIOS

Local port forwarding

En local forwarding la aplicación cliente está localizada en el cliente SSH.
El socket pasivo para la boca del túnel se abre local (en el host del cliente SSH)



Port forwarding

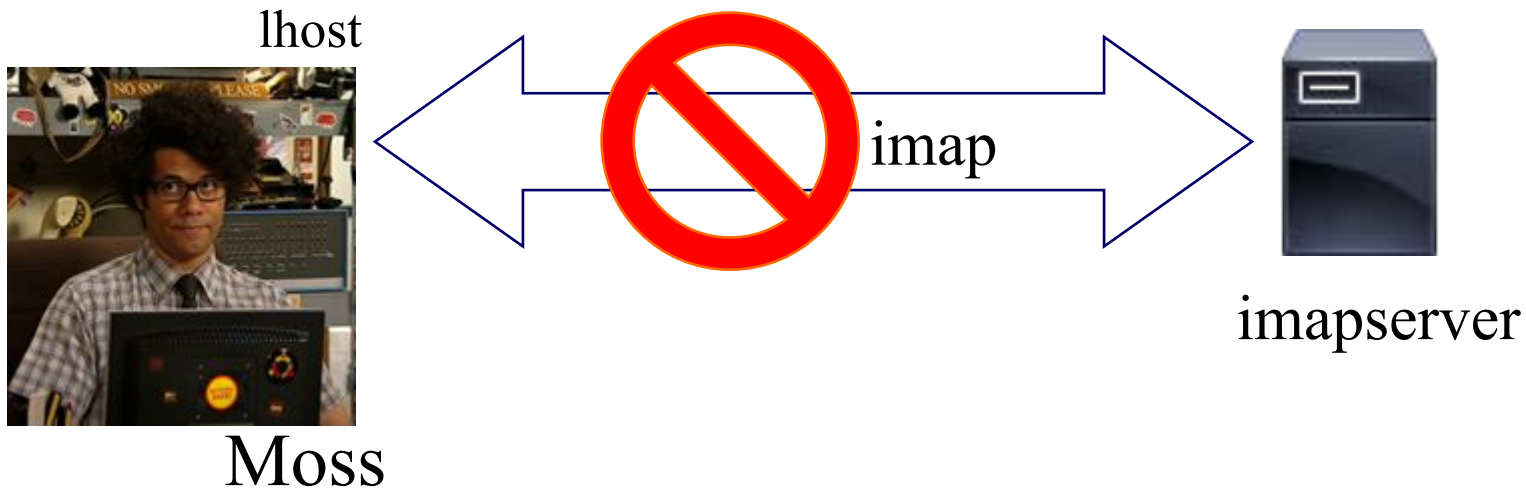
Local forwarding.

La sintaxis para indicar que se va a redirigir lo enviado a un puerto local a otro puerto en otro host es

```
$ ssh -Lport:host:hostport SSHserver
```

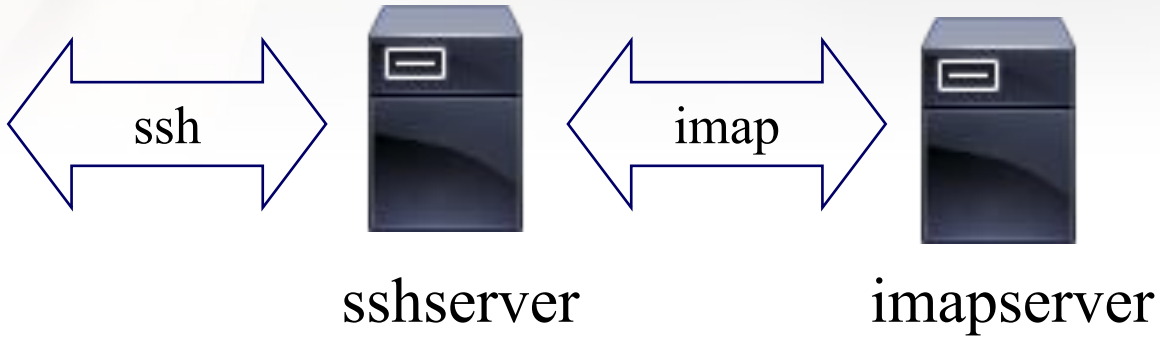
Local Port forwarding: ejemplo

Moss quiere conectarse a un servidor IMAP remoto pero por alguna razón no puede o no quiere hacerlo directamente



Local Port forwarding: ejemplo

Moss asegura la conexión IMAP con un servidor ssh remoto que pueda conectarse al servidor IMAP.



Paso 1: establecer la conexión

```
moss@lhost:~$ ssh -L2001:imapserver:143 -l user sshserver
```

Se establece una sesión remota con el servidor *sshserver* pero además hará que todo lo enviado al puerto local 2001 sea enviado a través de la sesión SSH al puerto 143 de localhost en *imaperver*.

Local Port forwarding: ejemplo

Paso 2: establecer propiedades en el cliente IMAP

Moss debe indicarle a su programa de correo que use como servidor localhost en puerto 2001.
Normalmente se conectaría al socket(imapserver,143), pero ahora debe hacerlo al socket(localhost,2001).

Local Port forwarding: ejemplo

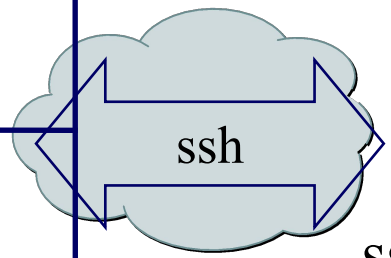
Ahora, los pasos para la conexión serán:

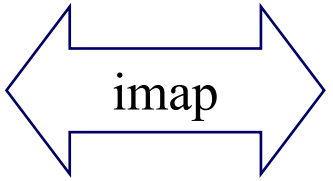
1. El lector de mails envía los datos a (localhost,2001)
2. El cliente local SSH lee el puerto 2001, encripta los datos, y los envía a través de la conexión SSH al servidor *sshserver*.
3. El servidor SSH desencripta los datos y los envía al servidor IMAP en el puerto 143 de *imapserver*.
4. Las respuestas son enviadas por el mismo túnel.

¿Qué debería cambiar si el servidor IMAP funciona en el mismo host que el servidor SSH?

Local Port forwarding: gateway

Los otros hosts de su red también quieren usar un túnel seguro a imapserver





sshcliente

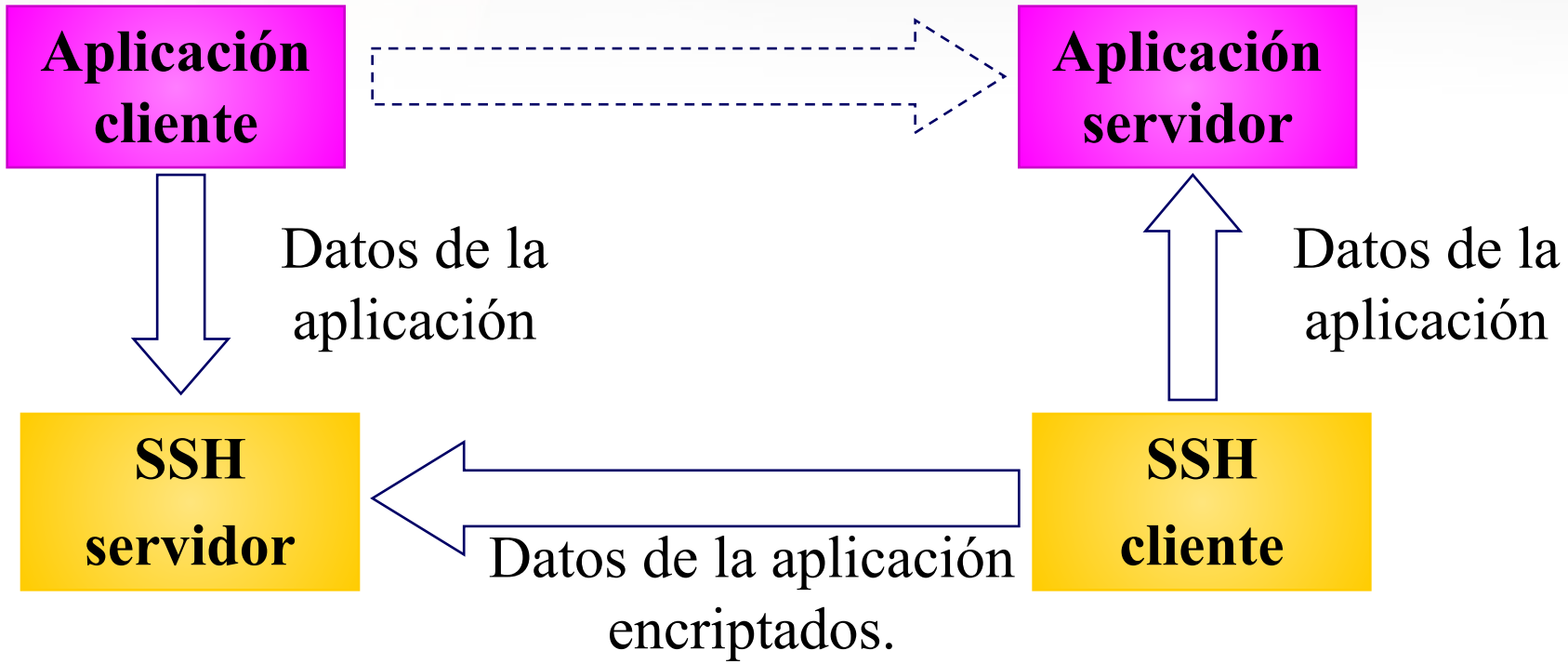
sshserver

imapserver

```
sshcliente:~$ ssh -g -L2001:imapserver:143 -l user sshserver
```

Remote port forwarding

En remote forwarding la aplicación cliente está localizada en el servidor SSH.



Port forwarding

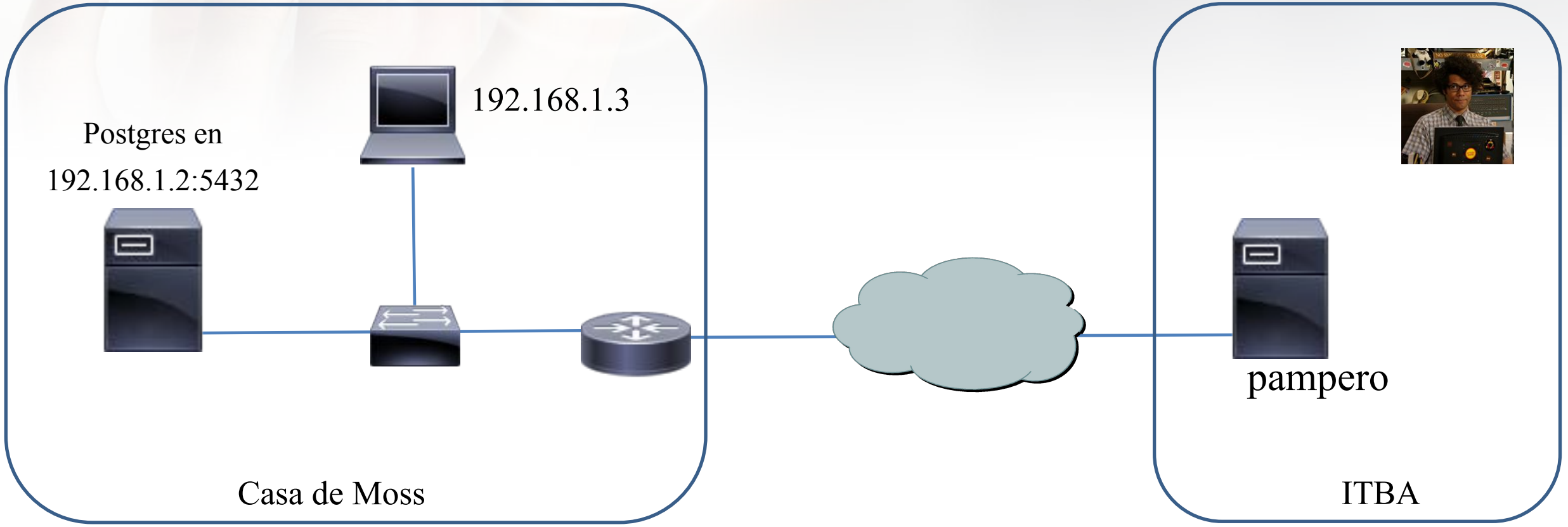
Remote forwarding.

Similar a local forwarding pero el cliente ssh es el que tiene acceso al servidor de la aplicación.

```
$ ssh -Rport:host:hostport sshServer
```

Remote Port forwarding: ejemplo

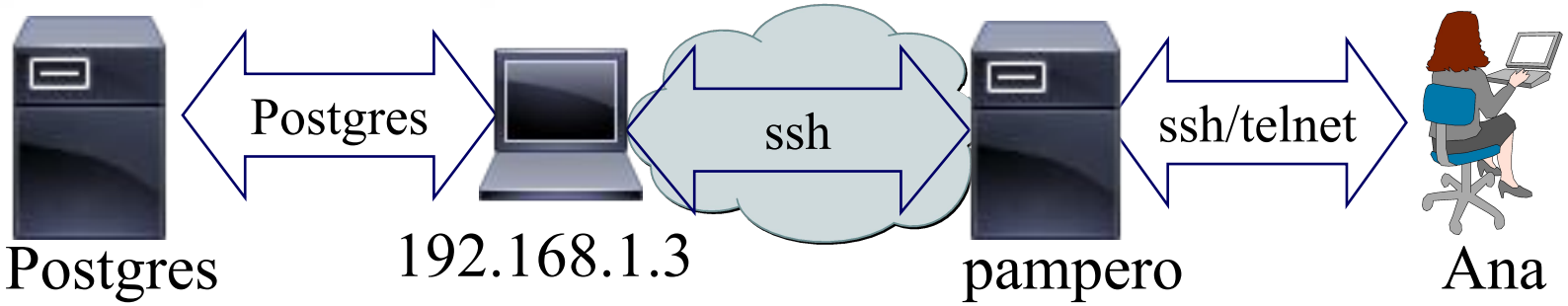
Moss tiene en su casa un servidor Postgres y quiere conectarse desde el ITBA.




```
192.168.1.3:~$ $ ssh -R2001:192.168.1.2:5432 moss@pampero
```

Port forwarding remoto

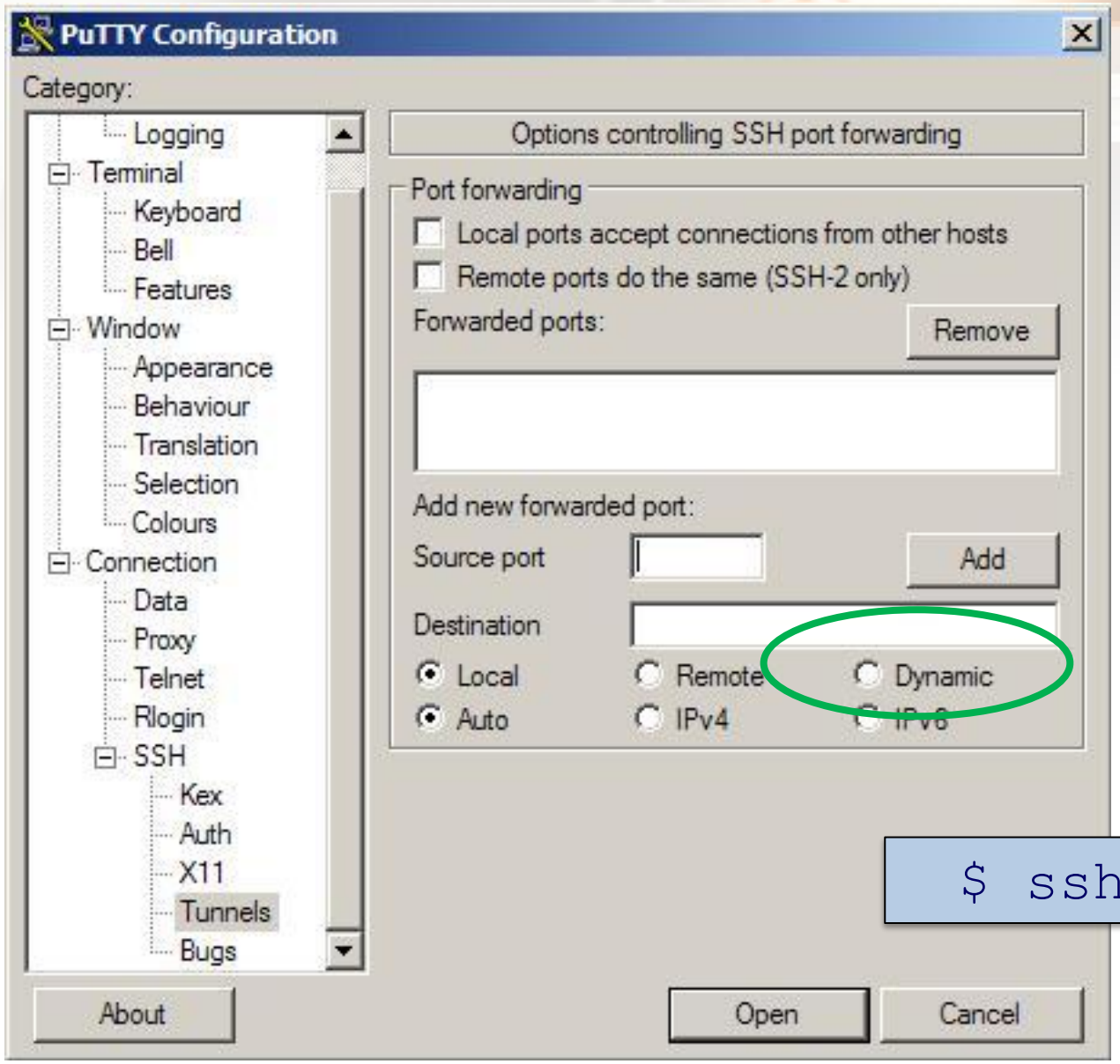
El puerto remoto será el 2001, el puerto local el 5432.
Una vez establecida la sesión, se ha creado un túnel desde el puerto 2001 del host remoto al puerto 5432 en el servidor Postgres.

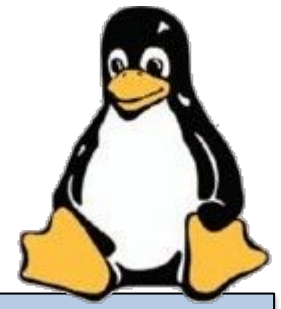



```
ana@pampero:~$ psql -p 2001
```

Por defecto todos los usuarios en pampero podrán usar el túnel

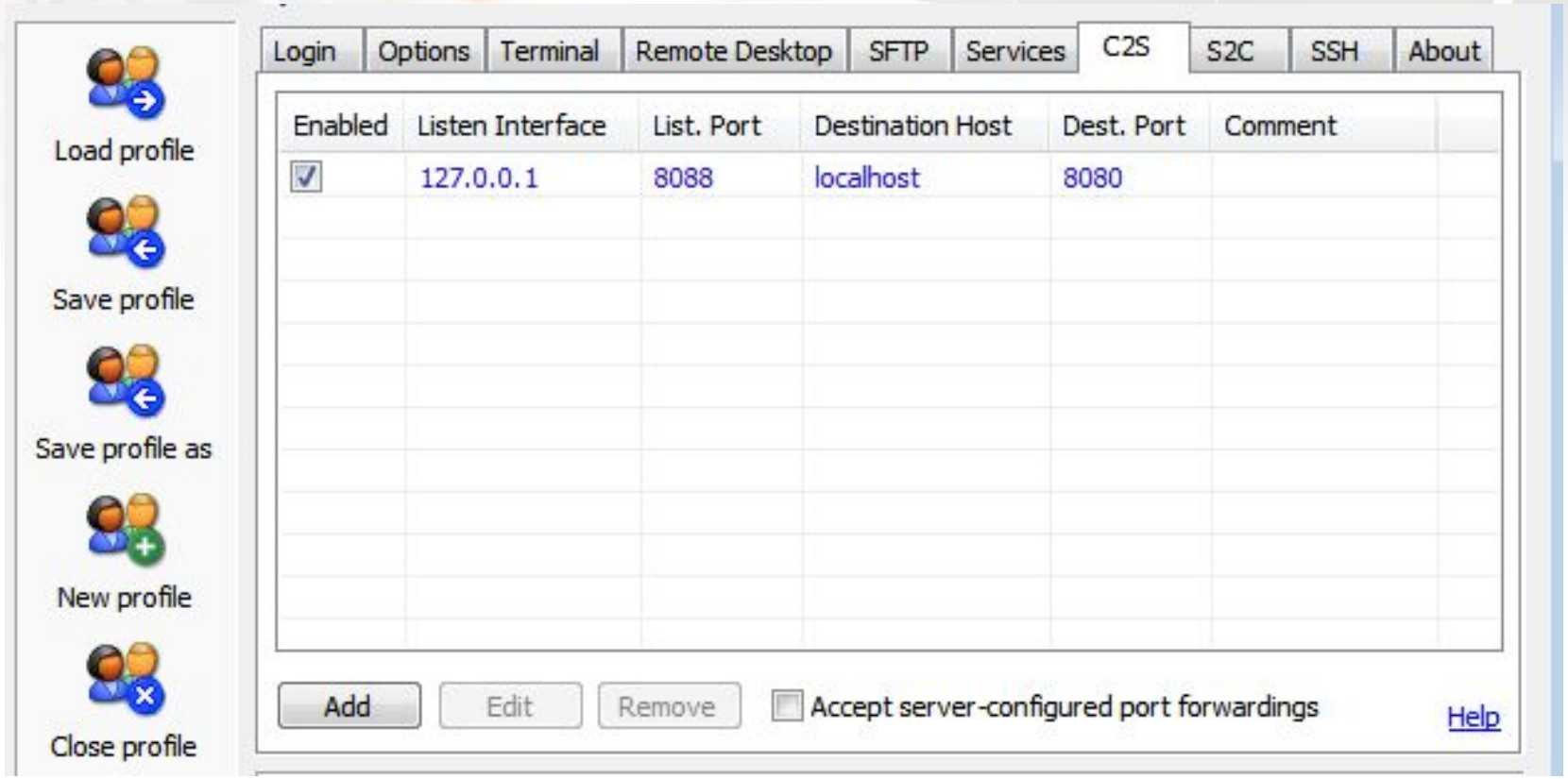
Dynamic Port forwarding



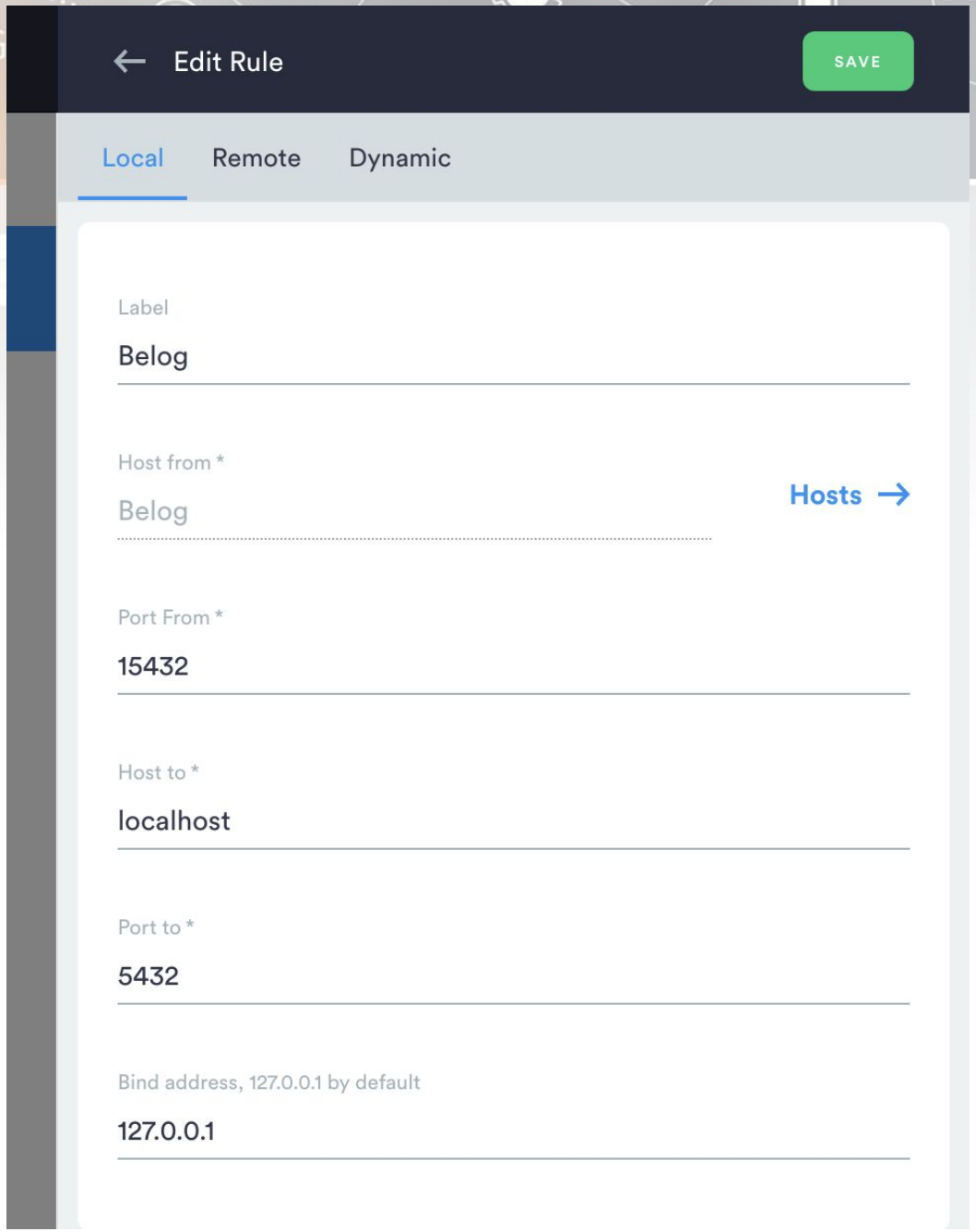


```
$ ssh -D 1080 pampero.itba.edu.ar
```

Port forwarding: BitVise



Port forwarding: Termius



Autenticación

Existen tres formas de autenticar un cliente SSH:

- Claves
- Claves públicas
- Basada en el host

Autenticación por claves

Es la forma más común y controla que el nombre de usuario y clave sean válidos, generalmente usando el archivo /etc/passwd o /etc/shadow.

Para deshabilitar este método editar /etc/ssh/sshd_config

```
PasswordAuthentication  no
```

Autenticación

Claves públicas

Requiere que cada usuario genere un archivo con su clave pública y otro con su clave privada.
La clave pública debe ser copiada al servidor y la clave privada debe estar en el host cliente.

En el archivo sshd_config debe figurar

```
PubkeyAuthentication yes
AuthorizedKeysFile  .ssh/authorized_keys
```

Autenticación

Pasos a seguir en el host cliente Linux

```
user@server:~$ ssh-keygen -d
Generating public/private dsa key pair.
Enter file in which to save the key
   (/home/.../.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_dsa
Your public key has been saved in id_ds.pub
```

Ahora el usuario dispone de archivos con clave pública y clave privada. Debe conservar la clave privada en el o los hosts origen y almacenar la clave pública en el o los hosts destino.

Autenticación

Pasos a seguir en el host servidor: **Copiar el archivo .pub**

```
user@server:~$ cd
user@server:~$ cat id_dsa.pub >>.ssh/authorized_keys
user@server:~$ rm id_dsa.pub
```

Autenticación

Usar clave pública con Putty y OpenSSH

1. Ejecutar el programa puttygen para generar las claves en el host cliente
2. Indicar en Putty, dentro de la opción Connection / SSH / Auth el archivo que contiene la clave privada
3. Agregar la clave en el servidor, de la misma forma que en el ejemplo anterior.

En caso de poseer archivos de clave privada generada por ejemplo con OpenSSH se puede utilizar puttygen para importar dicha clave.

Autenticación

El cliente debe autenticar al servidor (*Server authentication*).

Soy Moss (*primera conexión*)

Clave pública de pampero

Pampero

Soy Moss

mensaje cifrado con clave privada pampero

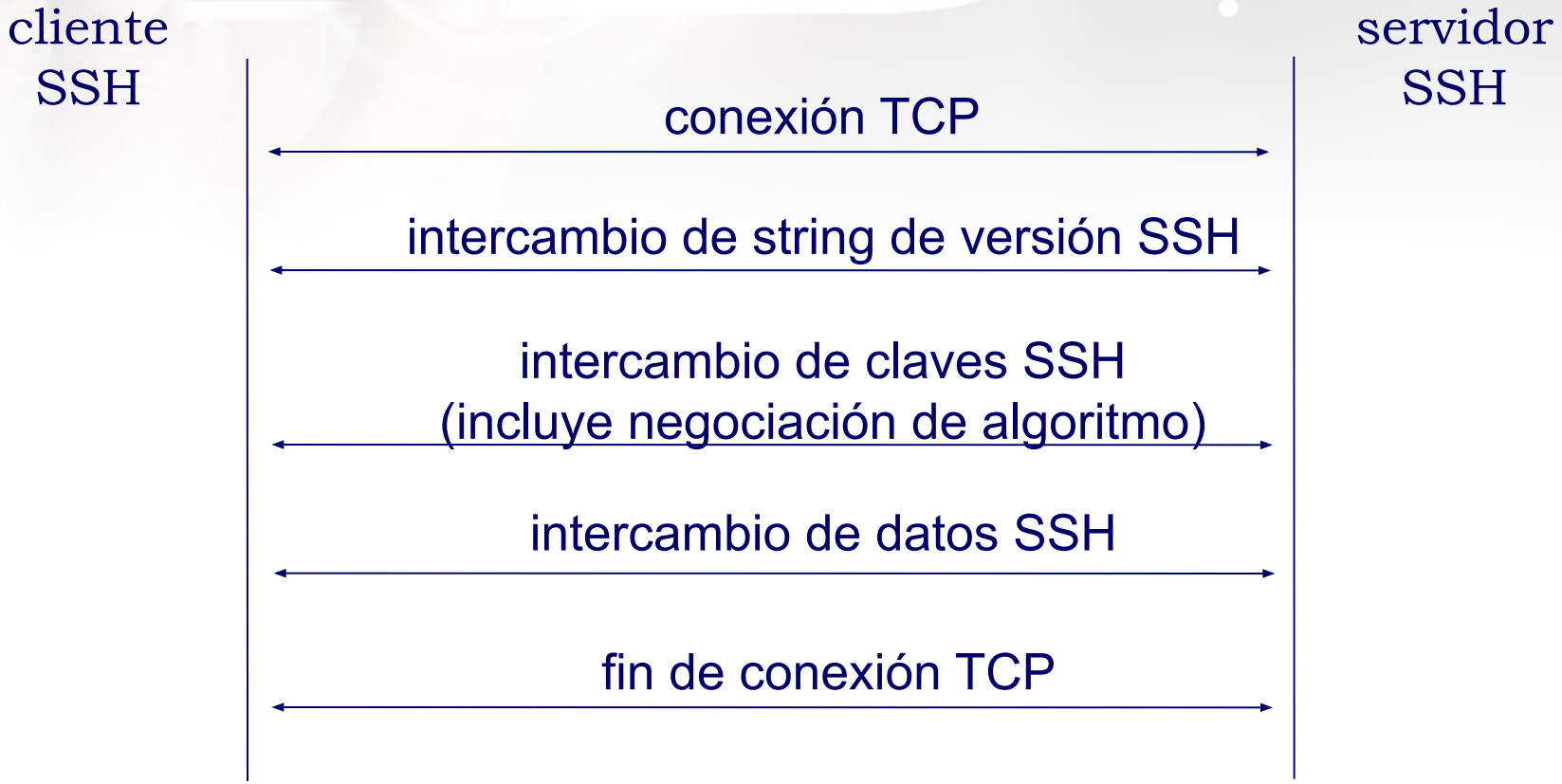
Modelo de capas SSH

Aplicación SSH	
Autenticación de usuario	Manejador de Conexiones
Transporte (*SSH TLP*)	
TCP	

Modelo de capas SSH

- Transporte (SSH Transport Layer Protocol)
  - Provee autenticación de servidor, privacidad e integridad
  - Opcionalmente puede comprimir datos
  - Utiliza los servicios de TCP
- Autenticación (SSH User Authentication Protocol)
  - Provee autenticación del lado del cliente
- Conexión (SSH Connection Protocol)
  - Se encarga de multiplexar el túnel seguro en los distintos canales lógicos
  - Algunos canales lógicos posibles
    - sesiones de shell seguras
    - TCP port forwarding
    - Conexiones X11

Conexión SSH



Material de lectura

https://www.climagic.org/tutorials/SSH_Tutorial_for_Linux.php