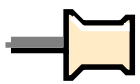
Acceso remoto a servidores

Álvaro González Sotillo

14 de enero de 2016



1. Introducción
2. Ejecución de programas
   - Telnet
   - SSH
3. Transferencia de ficheros
   - FTP
   - SCP
   - SFTP

Introducción

- En entornos pedagógicos, es común tener acceso directo a la consola (hardware) de los servidores
  - Virtual Box / VMWare Player
  - XAMPP
- En un entorno empresarial suele haber varios entornos
  - **Desarrollo**: De fácil acceso y modificación
  - **Producción**: Acceso físico protegido, accesible mediante red

Operaciones

- Son necesarias las siguientes operaciones sobre servidores remotos
  - **Ejecución de programas**: Arranque/parada de servidores, monitorización, modificación de ficheros, instalación de programas...
  - **Transferencia de ficheros**: Programas, HTML, CSS, PHP, JS, imágenes...

Ejecución de programas

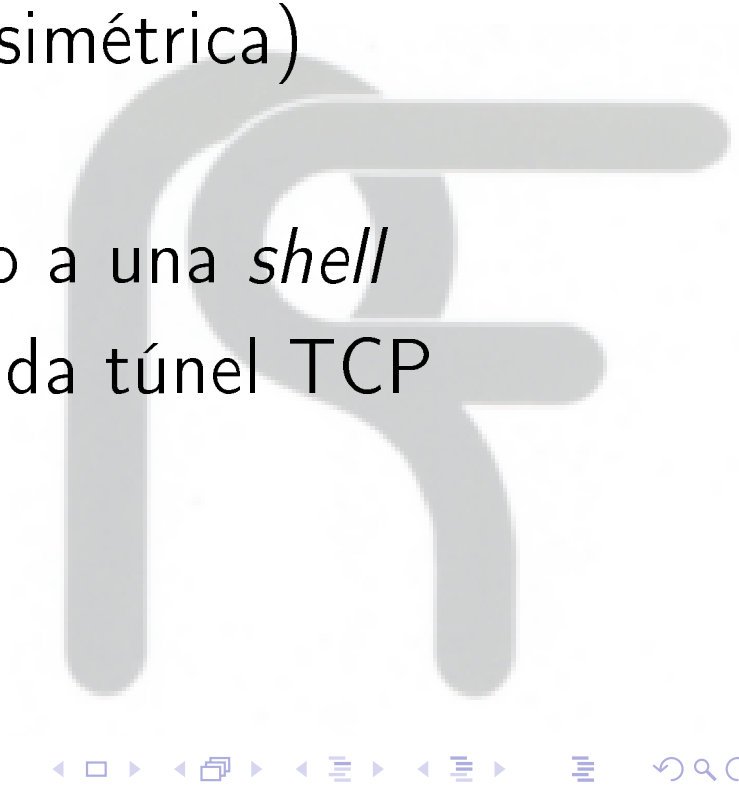
- Modo gráfico
  - Consumen más recursos de los sistemas remoto y local (memoria, CPU, red)
  - Pero són mucho más *cómodos*
  - VNC: Simple, portable. Sólo comparte pantalla, teclado y ratón
  - RDP: Más usado en Windows. Puede compartir sonido, impresoras y unidades de disco.
- Modo texto (línea de comandos)
  - Telnet
  - SSH

Telnet

- Conexión a *shell* remota
- Sin seguridad (la contraseña se envía como texto plano)
- Se utiliza aún en algunos dispositivos (routers ADSL, Switches...)

SSH

- *Evolución* de Telnet
- Conexión segura (criptografía asimétrica)
- Varios canales de comunicación
- Por defecto, un canal conectado a una *shell*
- Opcionalmente, canales para cada túnel TCP



Clientes SSH

- Linux
  - Comando `ssh` de **openssh**: Se puede utilizar desde cualquier terminal
- Windows
  - **Putty**: Incorpora un terminal y el cliente SSH.
  - **Kitty**: Basado en **Putty**, añade nuevas funcionalidades para Windows.
  - **openssh**: Versiones nativas o sobre **CygWin**

`https://en.wikipedia.org/wiki/Comparison_of_SSH_clients`

Servidores SSH

Los más comunes son:

- **Openssh**: Servidor por defecto en *Linux*
- **FreeSSHd**: Para Windows

`https://en.wikipedia.org/wiki/Comparison_of_SSH_servers`

OpenSSH Server

- Durante la instalación se crea una clave asimétrica para identificar al servidor
- Tras la instalación, puede interesar que el usuario `root` pueda acceder con contraseña
  - Fichero `/etc/ssh/sshd_config`, opción `PermitRootLogin` a `true`

```
$ apt-get install openssh-server
....
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
Creating SSH2 ECDSA key; this may take some time ...
....
```

Listado 1: Instalación de openssh-server

Autentificación SSH: Servidor

- En la primera conexión, se debe validar la clave pública enviada por el servidor
  - Es responsabilidad del usuario validar el hash de la clave pública por otros medios
- El fichero `~/.ssh/known_hosts` contiene una línea por cada servidor validado

```
$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 9b:1c:2e:47:5f:2c:18:b5:0d:62:01:3c:f0:ab:8e:70.
Are you sure you want to continue connecting (yes/no)?
```

Listado 2: 1ª conexión SSH

Autentificación SSH: Servidor

- Si cambia el nombre, IP o clave del servidor, se mostrará un mensaje de error
- El fichero `~/.ssh/known_hosts` deberá ser modificado a mano

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
6e:45:f9:a8:af:38:3d:a1:a5:c7:76:1d:02:f8:77:00.
Please contact your system administrator.
Add correct host key in /home/XX/.ssh/known_hosts to get rid of this message.
Offending RSA key in /var/lib/sss/pubconf/known_hosts:4
RSA host key for pong has changed and you have requested strict checking.
Host key verification failed.
```

Listado 3: Autenticidad sospechosa de un servidor SSH

Autentificación SSH: Cliente

SSH permite más de un método de autentificación

- **Contraseña**
  - De un usuario del sistema (o virtual)
- **Clave asimétrica**
  - Se crea con ssh-keygen
  - La clave pública creada (`.pub`) se debe llevar a la máquina remota, añadiéndola al fichero `~/.ssh/authorized_keys`
  - El comando ssh-copy-id puede llevar la clave por nosotros si tenemos acceso por contraseña

Autentificación SSH: Cliente

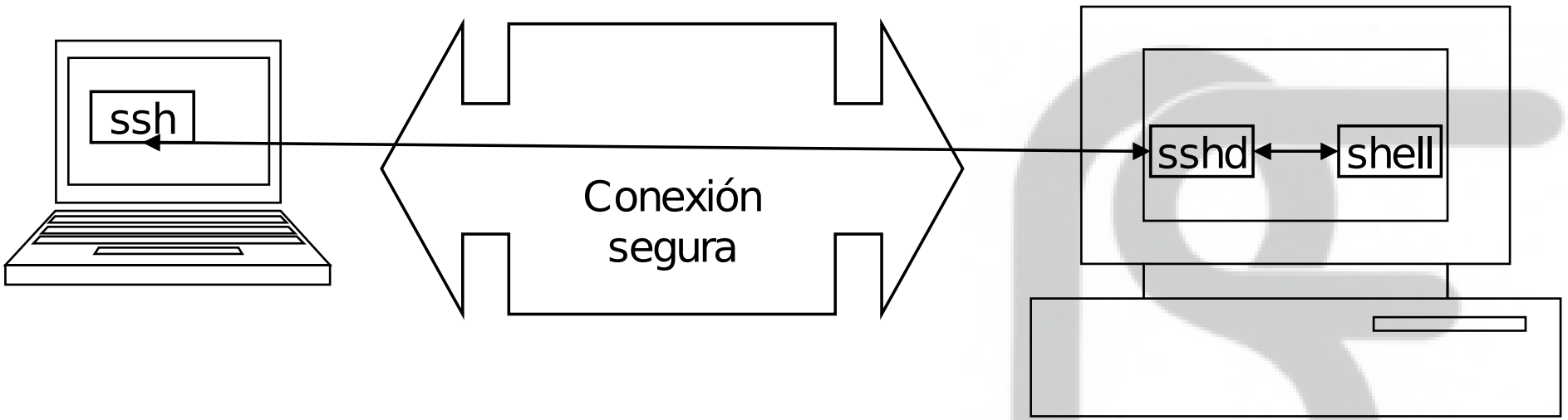
Conexión a *Koding.com*

Sigue las instrucciones de http://www.koding.com/docs/ssh-into-your-vm para conectarte a tu máquina virtual de Koding por SSH.

Nota: En clase, debido al proxy, no puede realizarse este ejercicio

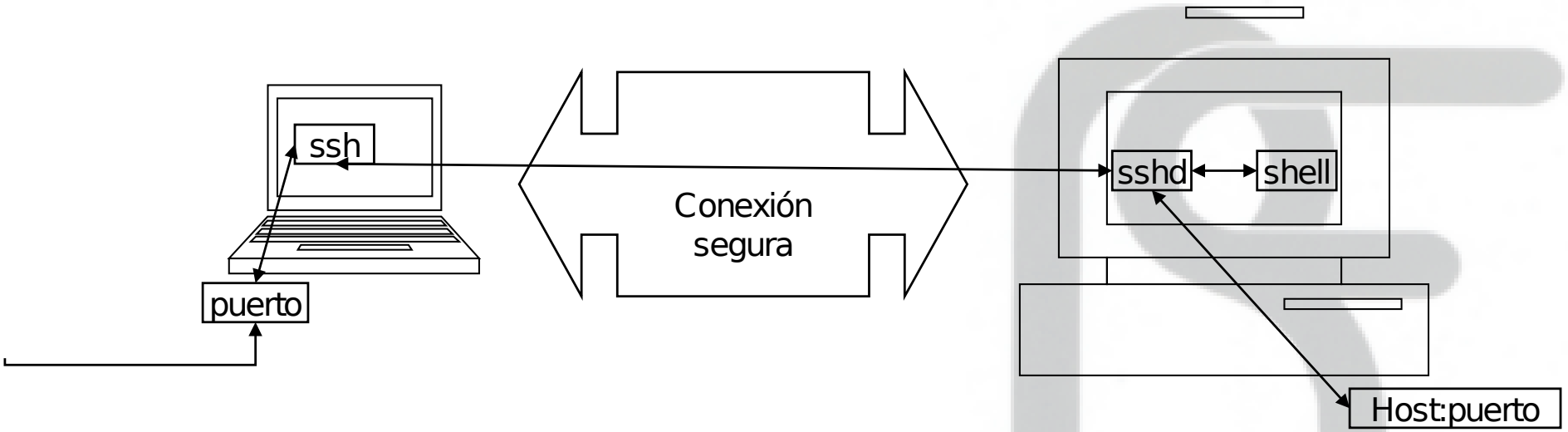
Túneles SSH

```
ssh usuario@servidor
```



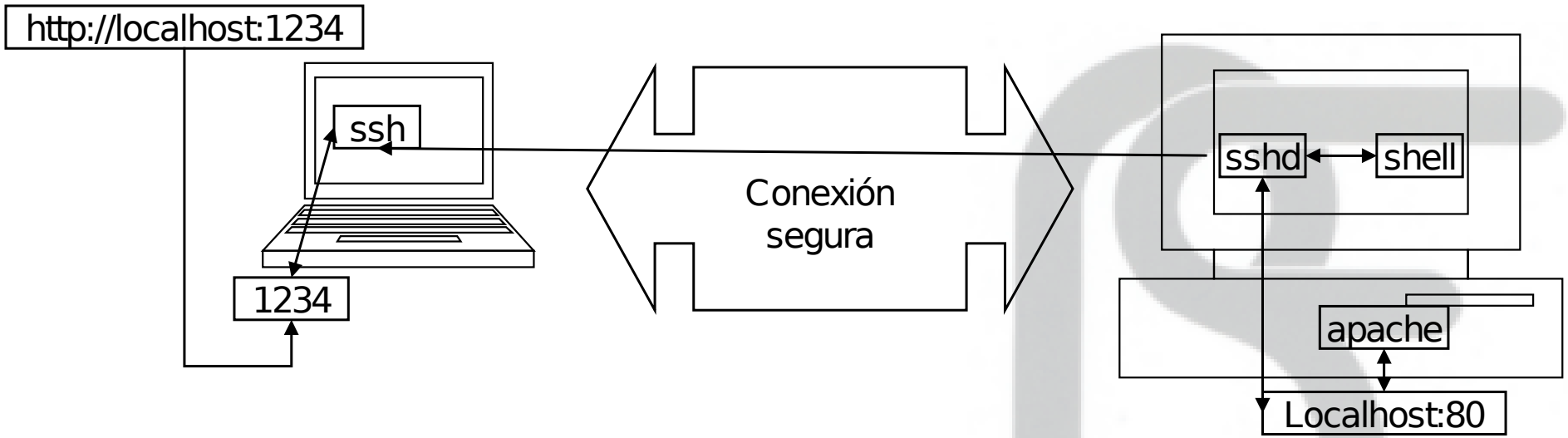
Túneles SSH

```
ssh -Lpuertolocal:hostdestino:puertodestino
usuario@servidor
```



Túneles SSH

```
ssh -L1234:localhost:80 usuario@servidor
```



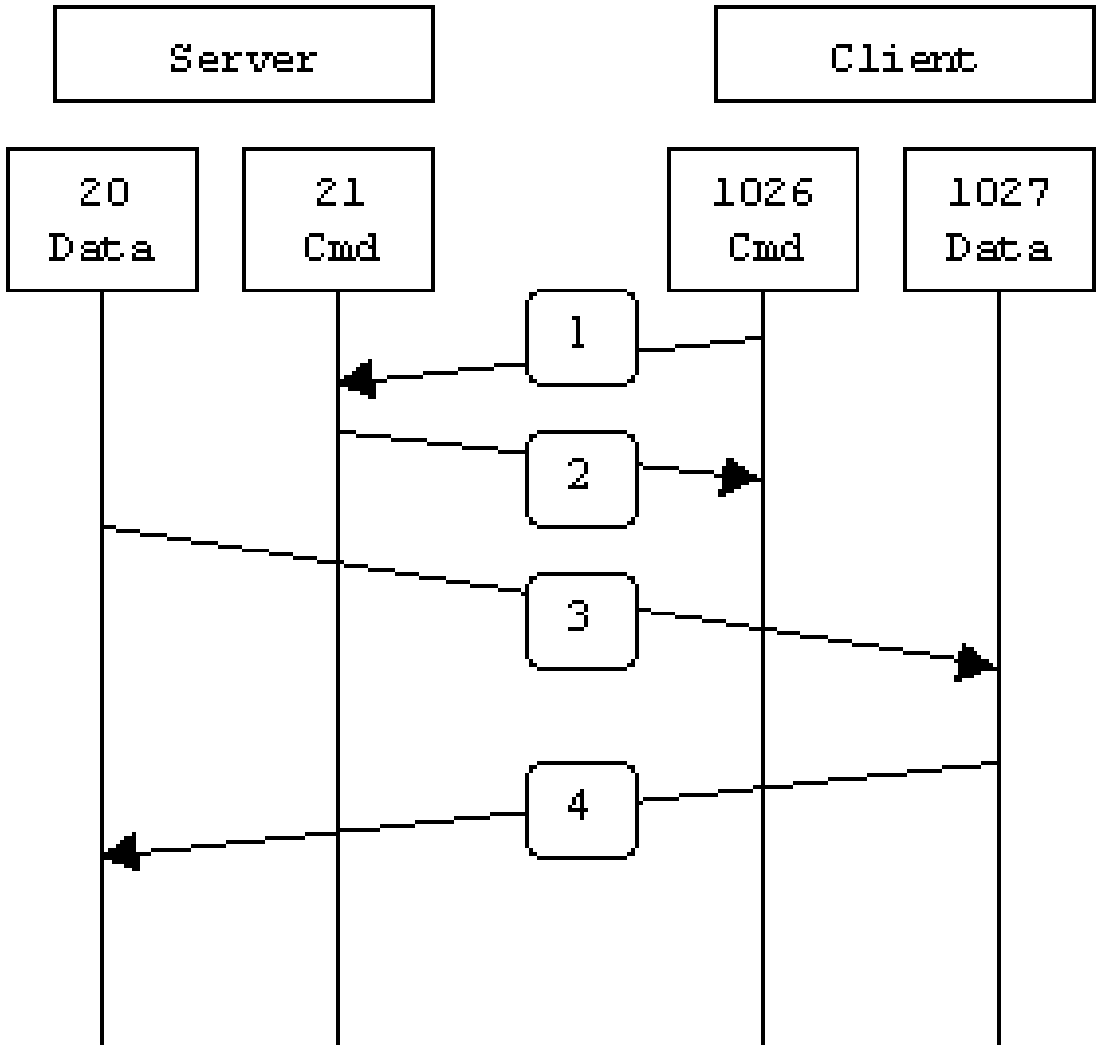
Transferencia de ficheros

- Los sitios web se desarrollan en local
- Para su despliegue en producción, se deben mover los ficheros del sitio al servidor final
- El protocolo más utilizado suele ser FTP (*File Transfer Protocol*)

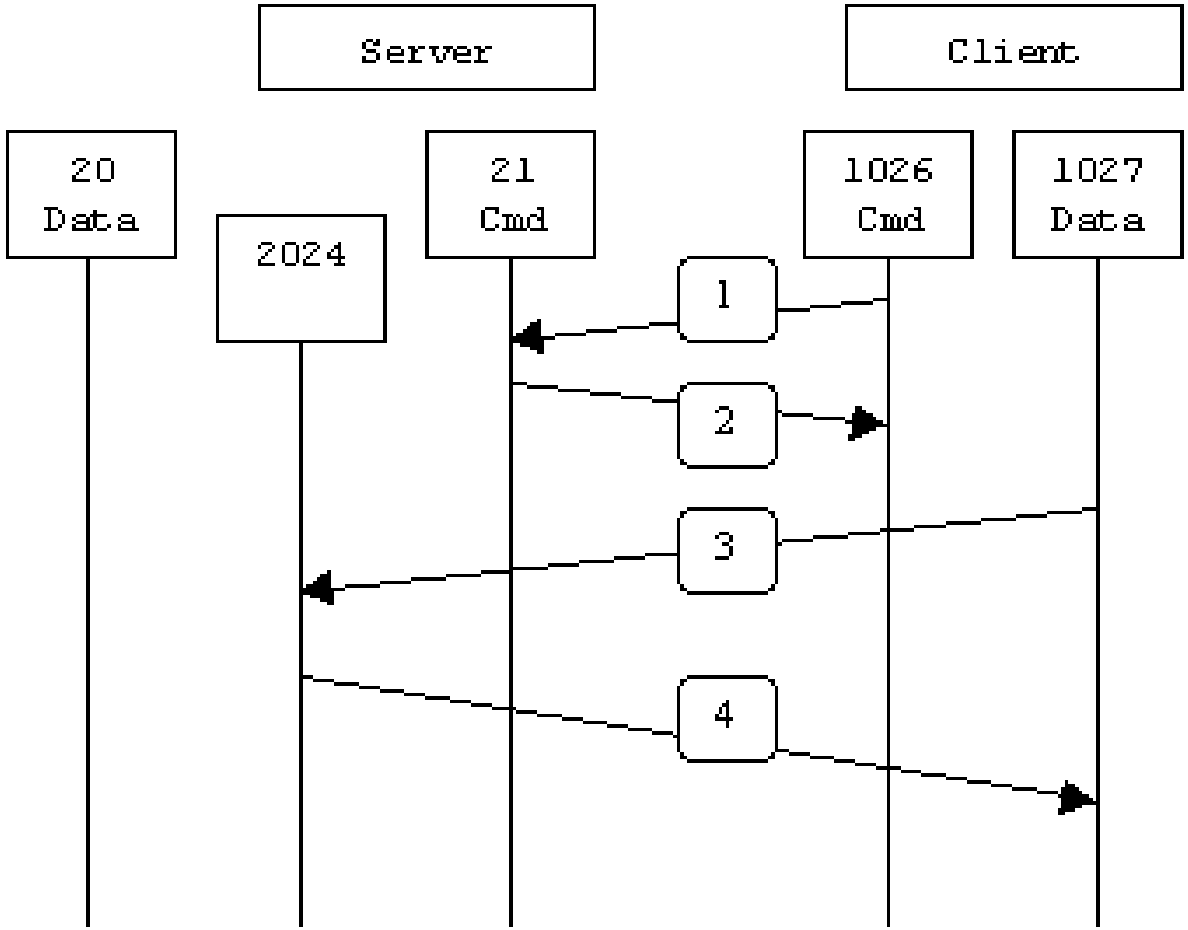
Protocolo FTP

- *File Transfer Protocol*
- Diseñado para subir o bajar ficheros de un sistema remoto
- Dos conexiones
  - Puerto 21: Conexión de control (órdenes, códigos de error)
    - Del cliente al servidor
  - Puerto 20: Conexión de datos
    - **Modo pasivo**: Del cliente al servidor (preferible para los *firewalls*)
    - **Modo activo**: Del servidor al cliente (por defecto)
    - Ver `http://www.slacksite.com/other/ftp.html`

FTP Activo

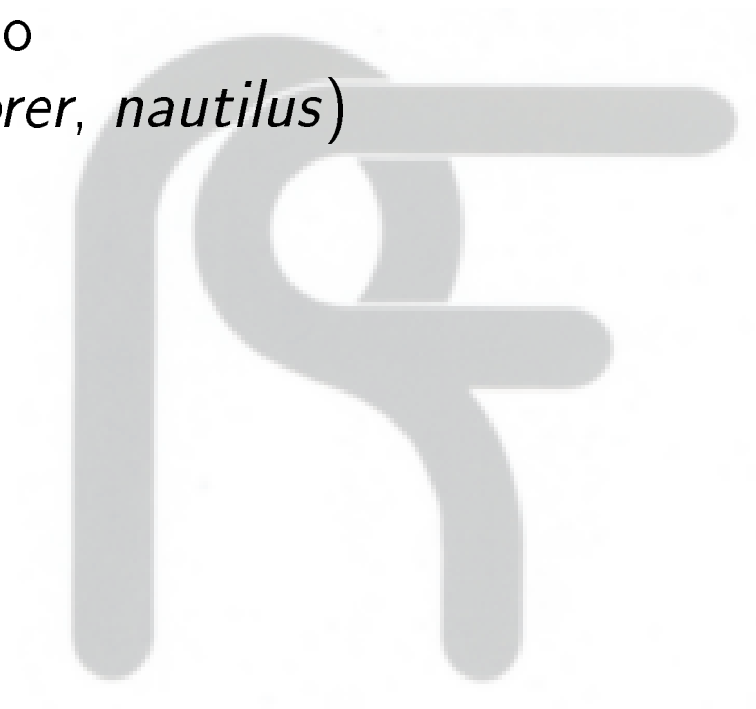


FTP Pasivo



Clientes FTP

- **Modo gráfico**
  - *Filezilla client*: Cliente avanzado
  - Exploradores de ficheros (*explorer*, *nautilus*)
  - Navegadores web
- **Modo texto**: Comando `ftp`



Comando `ftp`

- Disponible en *Windows* y *Linux*
- Los comandos se inspiran en los de la shell de **nix*:
  - `open`: Conexión a un servidor
  - `quit`: Finalización de la sesión FTP
  - `cd`: Cambiar el directorio remoto
  - `dir`: Listar el directorio remoto
  - `pwd`: Conocer el directorio remoto actual
  - `put`: Subir un fichero (local a remoto)
  - `get`: Bajar un fichero (remoto a local)
  - `mput`: Como `put`, con *wildcards*
  - `mget`: Como `get`, con *wildcards*
  - `bin`: Transferencias binarias
  - `!`*comando*: Comando ejecutado localmente

Transferencias binarias/texto

- El protocolo FTP se diseñó para transferencias de ficheros entre diferentes sistemas (*Windows*, *Unix*, *Mac*, *mainframes*...)
- Cada uno de estos sistemas utiliza un encoding/codepage distinto
- **Modo texto**: Cliente y servidor FTP traducen automáticamente el texto de los ficheros
- **Modo binario**: Se transmiten bytes sin traducir
- En la actualidad, se utiliza modo binario de forma generalizada

Ejercicios

Conexión a `ftp://ftp.microsoft.com`

Desde *linux*: conéctate al servidor `ftp.microsoft.com` y baja un fichero de texto mediante la línea de comandos (puede haber problemas con el proxy).
Renombra el fichero a `bajado-texto.txt`.
Baja de nuevo el mismo fichero, pero esta vez en modo binario.
Compara los dos ficheros (puedes utilizar el comando `diff`, o visualizarlos en binario con `xxd`).

Ejercicios

Conexión a `ftp://ftp.microsoft.com` con *Nautilus*

Desde *Nautilus*: conéctate al servidor `ftp.microsoft.com` (puede haber problemas con el proxy)

Comprueba como se puede navegar por la estructura de ficheros como si se tratase de ficheros locales.

Baja algún fichero del servidor (por ejemplo, copiando y pegando en una carpeta local)

Ejercicios

Conexión a `ftp://ftp.microsoft.com` con *Explorer*

Desde *Explorer*: conéctate al servidor `ftp.microsoft.com` (puede haber problemas con el proxy)

Comprueba como se puede navegar por la estructura de ficheros como si se tratase de ficheros locales.

Baja algún fichero del servidor (por ejemplo, copiando y pegando en una carpeta local)

Ejercicios

Conexión a `ftp://ftp.microsoft.com` con *Firefox*

Desde *Firefox* u otro navegador web: conéctate al servidor `ftp.microsoft.com` (puede haber problemas con el proxy)
Comprueba como el navegador crea una página web a partir de los directorios del servidor.
Baja algún fichero del servidor.

Ejercicios

Conexión a `ftp://ftp.microsoft.com` con *Filezilla*

Desde *Filezila* u otro navegador web: conéctate al servidor `ftp.microsoft.com` (puede haber problemas con el proxy)
Comprueba como se pueden inspeccionar los comandos FTP de bajo nivel al navegar por los directorios y transferir ficheros.

Servidor FTP

- Existen varias implementaciones de servidores FTP
  - Plugins para *IIS*
  - *Filezilla Server*, para *Windows* y *Linux*
  - *VSFTPD* para *Linux*
- En los ejercicios usaremos *VSFTPD* como servidor.

VSFTPD

- Utiliza por defecto las cuentas de sistema operativo
  - Login y password
  - Directorio inicial
  - Permisos de acceso
- Instalación:
  - `http://www.liquidweb.com/kb/how-to-install-and-configure-vsftpd-on-ubuntu-14-04-lts/`
  - `chroot_local_user=NO`
  - `anonymous_enable=YES`

Ejercicio VSFTPD

Instalar y probar VSFTPD

Instala VSFTPD en tu servidor *Debian*. Comprueba su funcionamiento con el cliente *Filezilla*.

- Con el usuario **alumno**
- Con el usuario **anonymous**, y una cuenta de correo como contraseña

SCP

- Basado en **SSH**
- Comando similar a **cp**
  - `scp` *`origen destino`*
  - *origen* y/o **destino** pueden ser ficheros locales o remotos
  - **Local**: Ruta local absoluta o relativa
  - **Remoto**:
    `servidor@usuario:ruta-relativa-al-home`

Protocolo SFTP

- Parecido al protocolo **FTP**, con las siguientes ventajas
  - Nivel de seguridad similar a **SSH**: un servidor **SSH** suele ser servidor de **SFTP**
  - Configuración de red más simple: Sólo una conexión, en vez de una para control y otra para datos
- Inconvenientes frente a **FTP**
  - Mayor consumo de CPU: Encriptación
  - Menor configuración: Sin usuarios anónimos

Ampliación

- Sistema de ficheros SSHFS
- VPN
  - OpenVPN: http://www.redeszone.net/redes/openvpn/
  - VPN con SSH: http://www.vicente-navarro.com/blog/2010/11/05/vpn-con-openssh/
- Para saber más
  - Servicios de Red e Internet (ISBN: 978-84-1622-832-4) Editorial Garceta.