2º SMR. Servicios en Red

Bloque 1: Acceso remoto

Práctica 1.2
SSH(1)

Luis Garcia <lgarcia@ausiasmarch.net>
Sandra Villanueva <svillanueva@ausiasmarch.net>

Introducción

Para la práctica necesitaremos tres máquinas virtuales, aunque sólo necesitaremos tener encendidas dos de ellas cada vez.

La primera máquina es el servidor Ubuntu server instalado en la práctica 0, esta MV estará encendida durante toda la práctica.

También usaremos dos clientes, (que podemos arrancar alternativamente) uno con un Linux con entorno gráfico (lubuntu 14.04 en nuestro caso) y por último una MV con Windows (usaremos Windows 7, pero serviría cualquier versión indistintamente).

Configuración de red

Los tres equipos deben estar en la misma red y ser accesibles entre si. La configuración de red de las tres máquinas virtuales es la misma que en la práctica anterior:

- Crearemos dos tarjetas de red en cada MV
- La primera interfaz de red la configuramos para permitir el acceso a internet. La configuración cambia si estamos trabajando en el aula del instituto o en casa:
  - GRUPO PRESENCIAL: Conectaremos la primera tarjeta de red en modo puente sobre eth0 para obtener una dirección de la red del aula.
  - GRUPO SEMIPRESENCIAL: Si no estamos en la red del aula, conectaremos la tarjeta en modo NAT.
- la segunda tarjeta la conectaremos a la red solo anfitrión que creamos en la práctica anterior (vboxnet0 en Linux)

Tanto en Windows como en el Network Manager del cliente lubuntu, dejaremos ambas tarjetas en modo automático (dhcp).

El fichero /etc/network/interfaces del ubuntu server, será similar a:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
      address 172.20.100.1
      netmask 255.255.255.0
```

Nota: En el guión de la práctica usaremos como IP del servidor el valor indicado (172.20.100.1) pero si la práctica se realiza en clase, el tercer octeto de la interfaz eth1 ("100") variará, y debemos sustituirlo dependiendo de nuestro número de PC (101, 102, ...)

Requisitos previos de software

En el **servidor Ubuntu**, debemos tener instalado el paquete openssh-server. Si no realizamos la instalación durante la práctica 0, podemos instalarlo ahora con el comando:

```
sudo apt-get install openssh-server
```

En el cliente Linux usaremos también el Wireshark en uno de los ejercicios, y en Windows necesitaremos el programa "putty", pero estas instalaciones se detallan más adelante.

Ejercicio 1: Conexión desde el cliente Linux

Arrancaremos las máquinas virtuales de ubuntu-server y el cliente Linux.

*Nota: Alternativamente, **si realizamos la práctica en clase, o nuestro anfitrión está ejecutando cualquier distribución de Linux, podemos sustituir el "cliente" por el propio anfitrión**, y realizar la práctica más cómodamente desde nuestro propio ordenador.*

Iniciaremos sesión en el cliente Linux con nuestro usuario de trabajo habitual ("ausias"). Abriremos un terminal, empezaremos comprobando la conectividad (verificaremos nuestra dirección IP y comprobaremos que podemos hacer ping al servidor).

Realizadas las comprobaciones básicas, nos conectaremos por ssh utilizando el cliente openssh, ejecutando el siguiente comando en un terminal del cliente Linux:

```
ssh -l "USUARIO" "IP-SERVIDOR"
```

o en la forma alternativa más popular:

```
ssh "USUARIO"@"IP-SERVIDOR"
```

Donde "USUARIO" e "IP-SERVIDOR" deben escribirse sin comillas y sustituirse por los valores apropiados del usuario y la ip del servidor, por ejemplo:

```
ssh ausias@172.20.100.1
```

Terminamos la conexión, escribiendo "exit" y regresamos al terminal de nuestro equipo.
Repite la conexión de nuevo y contesta a las siguientes preguntas:

Ejercicio 1.1 **Cuando nos conectamos por primera vez a un servidor se muestra un mensaje que nos obliga a confirmar que realmente queremos realizar la conexión.**
¿Por qué nos pide esta verificación? Por que va a guardar la huella digital en el registro y se le asignara a esa ip.
¿Qué ocurre el resto de veces que repetimos la conexión al mismo servidor? A que esta accion solo se lleva a cabo una vez.
¿A qué se debe este cambio de comportamiento? Que comprueba que. conrresponden la huella digital enviada con la almacenada.

Ejercicio 1.2 Ejecuta el comando "netstat -atun" tanto en el cliente como en el servidor (aprovecha la propia conexión ssh!). Muestra el resultado del comando y señala las líneas correspondientes a la conexión ssh en cada lado. ¿Qué criterio has usado para identificarlas?

He buscado el puerto 22 que es el usado por el ssh

```
ausias@sri-ubu-server:~$ netstat -atun
Conexiones activas de Internet (servidores y establecidos)
Proto  Recib Enviad Dirección local         Dirección remota       Estado
tcp        0      0 0.0.0.0:22              0.0.0.0:*              ESCUCHAR
tcp        0      0 172.20.102.250:22       172.20.102.1:48282     ESTABLECIDO
tcp6       0      0 :::22                   :::*                   ESCUCHAR
ausias@sri-ubu-server:~$
```

```
a026761180j@S04-PC19:~/a026761180-pool$ netstat -atun | grep :22
tcp        0      0 0.0.0.0:22              0.0.0.0:*              ESCUCHAR
tcp        0      0 172.20.102.1:48282      172.20.102.250:22      ESTABLECIDO
tcp6       0      0 ::22                    :::*                   ESCUCHAR
udp6       0      0 ::2250                  :::*
a026761180j@S04-PC19:~/a026761180-pool$
```

Ejercicio 2: Verificar la identidad del servidor

Una vez realizada una primera conexión, el cliente ssh guarda la clave pública de cada servidor al que nos conectamos. De esta forma, el cliente recuerda la "huella digital" de cada servidor, lo que permite realiza su comprobación en futuras conexiones.

Realizaremos el ejercicio con las mismas máquinas que en el anterior: ubuntu-server y el cliente Linux.

Eliminar el fichero de configuración del cliente "~/.ssh/known_hosts"

El cliente, almacena las firmas de los servidores en el HOME de cada usuario, dentro del fichero "known_hosts" en el subdirectorio oculto ".ssh".

Ejercicio 2.1. ¿Qué efecto tendría eliminar este fichero? Se perdera el listado de huellas digitales asignadas a cada ip de las conexiones ssh

Desde un terminal del cliente, borra el fichero con el comando:

```
rm ~/.ssh/known_hosts
```

Una vez eliminado el fichero, realiza ahora una nueva conexión ssh al servidor desde el cliente. ¿Qué pregunta hace el cliente ssh? La pregunta que hace cuando te conectas a un nuevo servidor.
¿Por qué?Por que se ha borrado el registro**.**

Ejercicio 2.2 Una vez realizada la conexión, termina la sesión ssh con el comando "exit" para volver al terminal del cliente. Comprueba si existe ahora el fichero "~/.ssh/known_hosts", si es legible su contenido y cuantas líneas tiene. Indica y justifica el resultado obtenido. El contenido solo es entendible por alguien que sepa la estructura de las huellas digitales

```
|1|bHnNI+zdVl2maNAeoW7/75oTgYo=|c3+6FORsXRYc+vNhzhzOrsU6vtE= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBPNfL9pmFS1LbJPUKn/upbjjv91o4y2aE2JHkQDjatQ3Ug6+GFOivvwlb94U5M4ciPbVrWhP81wTR34j9BOucEQ=
|1|+GCZMOcMa5daAYG90dfrXwQt4q4=|Em5twEJB/oUXMvszEyMfda2EVm8= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBDDSSS++cHVajJO2hxSWLC0jpJHKFb0IasLacOdE3LST3Vd7l50aCxxar/NpdjLoyf4i+lTtpp9H3I4NCKl+b8=
|1|QYL4qaGuFoCHvL39CbuaIHC3t4o=|Bi3zlS5AyLB8Js8luodK5qYy3/Q= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBFHq1rI+KxWdZjxH2aflaNh23P4BxIOD7QcQTOk3L8rsKRuTaVhdW9VJ2bXVPngfheMA15tCINspJdKF1zK2qbc=
|1|envicxoezkclS+c5bT8jEQSdPm0=|E6lJM/iocQ2VJLPeuHGns6n8UC8= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBAkzd/hicSWwKfBxDOD0L5Bu3b+o1D2enVQlq7QRTp8/2bLY18dkl1aC724ENJ+1LjVRYILDGwYJZgjiePlmEO8=
|1|THxS2/gl4hziuwCR8Rdju3jVo2A=|HtQP4K2ZcmAqc8MU/GPIZvYRtbc= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBGMl3POFBE2oIUfY0p9uMArcMX6Xk21azE/t5RFGfVS23NBs0P0m9fvmVw1wR8wAGW16XVQYcWhLQIk5bvIyxvc=
|1|ELy1NEhh8Ps5AjOUocwOVYC52IA=|nQT0Ly8ssh4IGDtxcQmHjBOge3g= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBOIa8gUVuMFY19IncBnb3J1dWC8nrP/duYLCRugRBzEcpGAUhMKLCx4oU45JUQxn290URmHYZ8iMHs/Kl8TmbUM=
|1|Rp3+X2IQXidBDLeoB7dYWMl0Yak=|gEKeOL9dcagFiHioyxeZEI8oVQY= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBIXWA9sH34HUgA5B0Vv+Tu3QOCC7ZNb3KAN7ycLIFSbwpe+F5EvGHTWUIPZFraQurvlpGHG2bFLdJI5ePjBN8+0=
|1|iRmCZHhciRCHl685Nzntfzjki/o=|fqSvPcDuDv5KBmBuThDkrCEGJsg= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBGVxcNF8/mESDnbB1U802L6PWyJbB3K4dDHF7ClbdZJqZrS1+2u6ZmVUK7+xSBT6PF0RBf003c5eynY0ts4nzIU=
|1|v7zVBl81ywiQ1yR8KPbQG4vKuzc=|k2yMbf4KdM92Rzv8h+v5eWvE1/g= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBIXWA9sH34HUgA5B0Vv+Tu3QOCC7ZNb3KAN7ycLIFSbwpe+F5EvGHTWUIPZFraQurvlpGHG2bFLdJI5ePjBN8+0=
|1|qntIzT1GfXb8CSFvpvOzrrzP52w=|bf3cKim6MTsvY3CxXZgHw5gW5bI= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBIXWA9sH34HUgA5B0Vv+Tu3QOCC7ZNb3KAN7ycLIFSbwpe+F5EvGHTWUIPZFraQurvlpGHG2bFLdJI5ePjBN8+0=
|1|FXegrENy/EO+t+Vl0q8YYaenRTk=|6EJC7ajKW2ml9a42VYT43jhwVuM= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBGVxcNF8/mESDnbB1U802L6PWyJbB3K4dDHF7ClbdZJqZrS1+2u6ZmVUK7+xSBT6PF0RBf003c5eynY0ts4nzIU=
```

Cambiar la huella del servidor

En condiciones normales, la clave pública de cada servidor no debería cambiar nunca, por lo que cualquier anomalía se interpreta como la señal de un posible ataque tipo "man in the middle" y el cliente no nos permitirá la conexión.
Vamos a forzar un cambio en la firma del servidor para ver como podemos actuar en estos casos. Las claves son aleatorias y se generan durante la instalación del servidor ssh, así que la forma
más sencilla para forzar la regeneración de las claves es desinstalar e instalar el paquete.
Accede a la consola de la máquina virtual Ubuntu server y ejecuta los comandos:

```
apt-get purge openssh-server
apt-get install openssh-server
```

Ejercicio 2.3 Vuelve al terminal del cliente y realiza de nuevo la conexión por ssh al servidor.

Indica que ocurre y que mensaje obtienes. ¿Has podido realizar la conexión? No, da error de que el host no corresponde.

Ejercicio 2.4 ¿Qué podemos hacer para que el cliente "vuelva a confiar" en el servidor? ***Indica al menos dos formas de resolverlo.***

(*AYUDA: por un lado, lee atentamente el mensaje que se muestra al intentar conectar …. Por otra parte, revisa con cuidado el desarrollo de este mismo ejercicio …*)

Debemos borrar la lista de host conocidos con:
`rm ~/.ssh/known_hosts`

ssh-keygen -R *(host)*

Ejercicio 3: Conexiones ssh desde Windows

Vamos a probar las conexiones desde el cliente Windows. Para este ejercicio, necesitamos arrancar el cliente Windows, si queremos ahorrar algo de memoria, podemos apagar el cliente Linux, que no será necesario ahora.

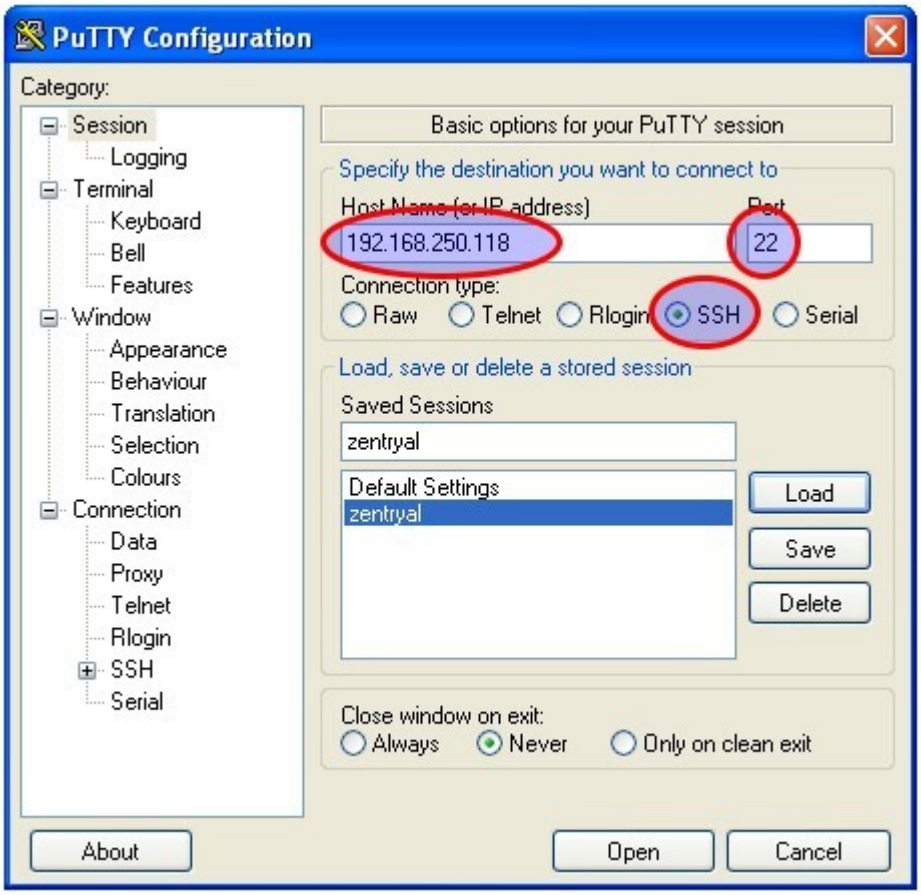
Instalación y configuración de PuTTY en Windows

PuTTY es un cliente ssh libre para Windows. Puede descargarse desde:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

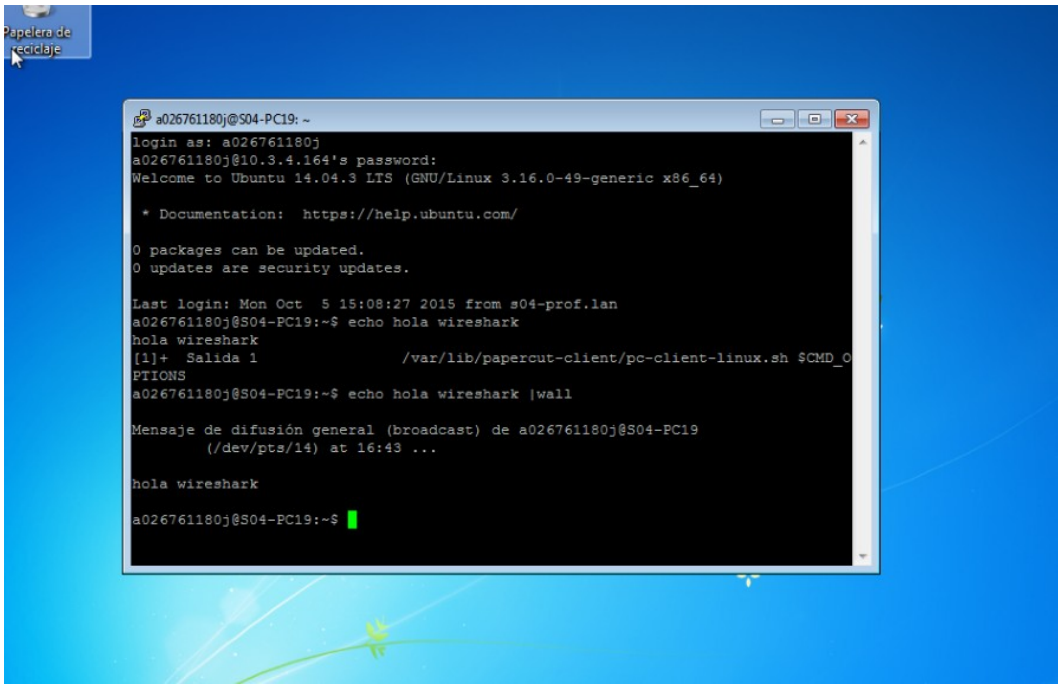
El uso básico de PuTTY consiste en introducir:

- el nombre o IP del servidor en el campo “Host Name”
- el Puerto en el campo Port (por defecto 22)
- Seleccionar 'SSH' en 'Connection type'.

A cada conexión se le puede dar un nombre y dejarla grabada. Posteriormente, podremos utilizarla de nuevo seleccionando su nombre en la lista de “Saved Sessions”.



Ejercicio 3.1 Introduce la IP del servidor y accede por ssh al mismo desde el cliente Windows usando putty. Adjunta una captura de pantalla.



Ejercicio 3.2 ¿Qué comando o comandos podrías utilizar mientras estás conectado por ssh para detectar la IP del cliente Windows desde el servidor, sin utilizar ningún comando ni utilidad de Windows? Razona la respuesta.

netstat -atun |grep :22 | egrep ESTABLECIDO

Netstat -atun muestra las conexiones y su puerto indicando origen y destino, si se filtra por :22 que es el puerto del ssh y otro filtrado por ESTABLECIDO aseguras que sea una conexión activa.

Ejercicio 4: Ventajas sobre telnet

Vamos a comparar el uso de ssh con su antiguo predecesor, telnet. Para ello, necesitamos primero instalar y configurar dicho servicio en el servidor Ubuntu, ya que, al no recomendarse su uso, no tiene sentido que venga instalado por defecto.
Realizaremos la práctica de nuevo con las máquinas virtuales ubuntu server y el cliente Linux por lo que podemos apagar el cliente Windows para ahorrar memoria.

Configuración del servidor

Desde un terminal de Ubuntu server (una sesión ssh es perfectamente válida), instalaremos los paquetes necesarios:

```
sudo apt-get update
sudo apt-get install openbsd-inetd telnetd
```

Instalando de esta forma, conseguiremos que el servicio telnet arranque "bajo demanda" cuando se produzca una conexión entrante.

Configuración del cliente

Para monitorizar la conexión, vamos a utilizar el programa Wireshark, que nos permite capturar tramas fácilmente desde el entorno gráfico del cliente Linux. Para instalarlo **en el cliente lubuntu, abrimos un terminal** y escribimos:

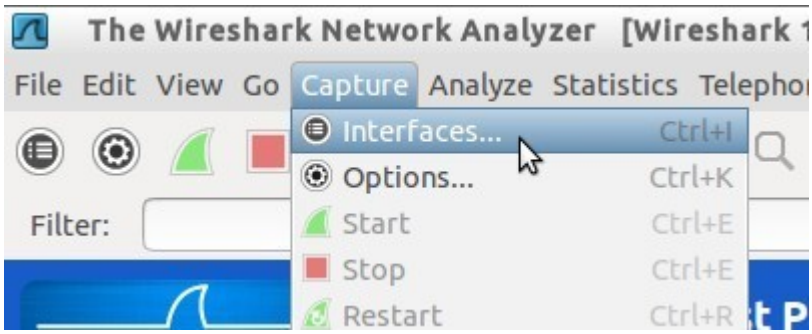
```
sudo apt-get update
sudo apt-get install Wireshark
```

Nota: si estamos usando el ordenador del aula como cliente, no necesitamos instalarlo, a que Wireshark viene ya instalado en los equipos aurex de clase.

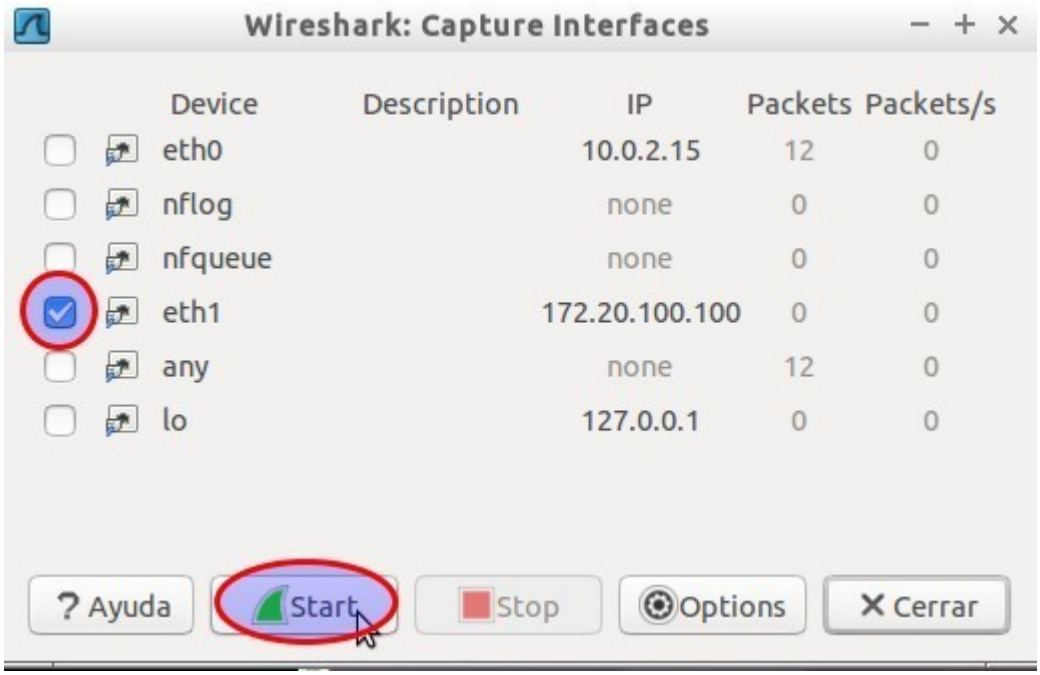
Una vez instalado Wireshark, lo lanzamos como root desde el terminal del propio cliente lubuntu:

```
sudo Wireshark
```

El programa mostrará un mensaje advirtiendo de los riesgos de ejecutarlo como root. Aceptamos y para iniciar la captura accedemos al menú:
Capture → Interfaces:



Buscamos la interfaz que conecta con el servidor. Si se trata de una máquina virtual cliente, la red solo anfitrión es la segunda tarjeta del cliente, es decir, eth1. Si estamos realizando la práctica desde el anfitrión, la interfaz en dicha red será vboxnet0.
Activamos la interfaz requerida e iniciamos la captura pulsando en "Start" :

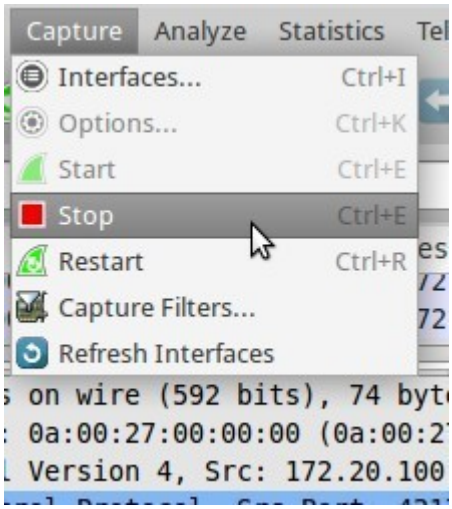


Dejamos en marcha el Wireshark y abrimos un nuevo terminal en el cliente.
Desde este nuevo terminal vamos a realizar al conexión por telnet al servidor, el comando necesario (ojo, en el aula tendrás que corregir la IP):

telnet -l ausias 172.20.100.1

Una vez introducido el password, conseguiremos nuestra sesión remota telnet en el servidor.
Ahora aprovecharemos para ejecutar algunos comandos sencillos a fin de generar algo de tráfico, y por último nos salimos de la sesión telnet con el comando "exit".

Si volvemos a la ventana del Wireshark, observaremos que la actividad ha sido registrada por el programa. Finalizamos la captura desde el menú Capture → Stop

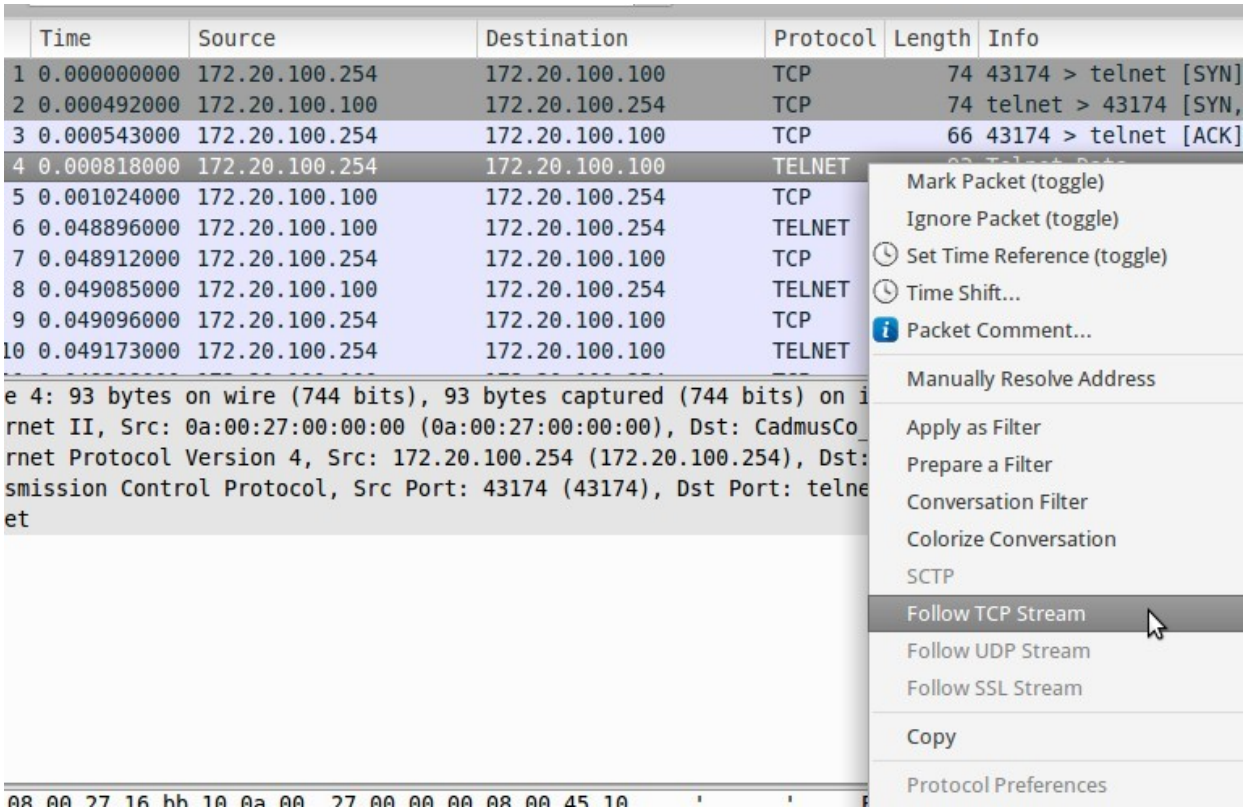


La ventana de Wireshark está dividida verticalmente en tres paneles que podemos redimensionar. En el de arriba observaremos la lista de paquetes capturados y su información básica (se indica el protocolo y algunas cabeceras) mientras que los otros dos paneles muestran detalles del paquete seleccionado.

Podremos ver el trafico telnet generado, y posiblemente algunos paquetes de otros protocolos.

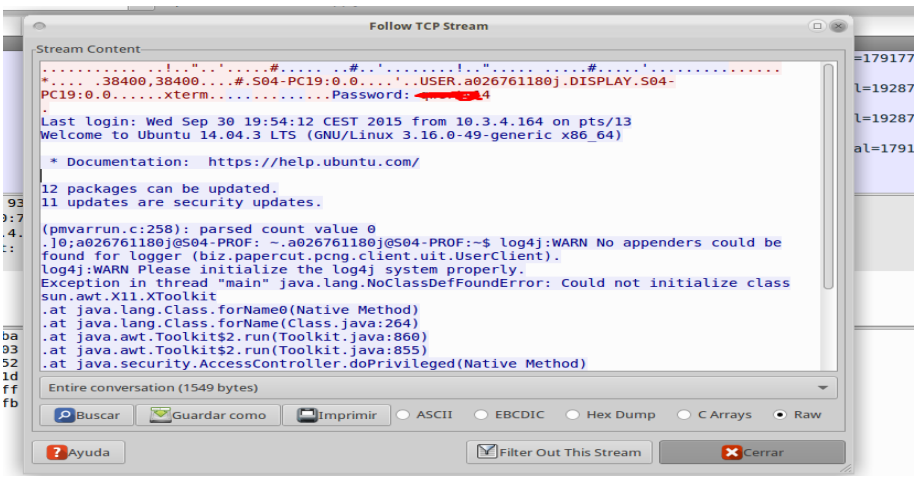
Vamos a pedirle a Wireshark que "junte" todas las tramas correspondientes a la conexión telnet para poder inspeccionar más cómodamente su contenido. Lo que haremos es usar las barras de scroll en el panel de arriba hasta visualizar el principio de la lista. **Debemos identificar el primero de todos los paquetes que se corresponda con el protocolo "TELNET".**

Una vez posicionados en el primer paquete TELNET, pulsaremos el botón derecho del ratón y seleccionaremos la opción "Follow TCP Stream", y el programa mostrará una ventana con todo el contenido de la comunicación.



Ejercicio 4.1. ¿Qué es lo puedes ver al unir el flujo de los paquetes telnet?
El contenido de la conexión
¿Es legible el contenido? Si
¿Se está utilizando algún tipo de cifrado? No
A la vista del resultado,¿qué conclusiones puedes extraer en términos de seguridad?.
Telnet es un servicio de conexión no seguro por que no esta encriptado.

Adjunta una captura que ilustre tus conclusiones y donde se observe el contenido de alguno de estos paquetes.



Vamos repetir ahora el proceso usando ssh. Sigue los pasos anteriores e inicia una nueva captura en Wireshark. Cuando el programa te pregunte si quieres guardar la captura anterior, contesta que no e inicia la captura nueva sin guardar la anterior.

Desde un terminal del cliente, conéctate por ssh al servidor y ejecuta una serie de comandos similares a los del ejercicio anterior.
Sal del ssh con "exit" , termina la captura en Wireshark y observa la lista de paquetes.

Encontrarás que la mayoría de paquetes tienen SSHv2 como protocolo. Sigue los mismos pasos que en el ejercicio anterior, localiza el primer paquete SSH y usa la opción "Follow TCP Stream", para seguir el contenido de la comunicación.

Ejercicio 4.2. ¿Qué es lo puedes ver ahora al unir el flujo de los paquetes ssh?
¿Es legible el contenido? No
¿Se está utilizando algún tipo de cifrado? Si
¿Puedes identificar los comandos que has usado durante la conexión? No
A la vista del resultado, ¿qué conclusiones puedes extraer desde el punto de vista de la seguridad al comparar telnet con SSH?.
Telnet, aunque mas rapido no tiene nigun tipo de seguridad a la hora de transmitir los datos con lo que es poco recomendable usarlo aparte que con las nuevas tecnologias la diferencias es dificil de apreciar con lo que SSH actualmente es mejor y sobre todo mas seguro

Adjunta una captura que ilustre tus conclusiones y donde se observe el contenido de alguno de estos paquetes.