Sesiones gráficas remotas

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

gsyc-profes (arroba) gsyc.urjc.es

Enero de 2018

Sesiones gráficas remotas

Definiciones

- Máquina local
  Equipo en el que trabaja el usuario, donde tiene su pantalla, teclado, y posiblemente, ratón
- Máquina remota
  Máquina donde se ejecuta la aplicación a usar. El usuario no tiene acceso a su teclado, pantalla ni ratón

Sesiones de texto / Sesiones remotas

- El protocolo ssh nos permite trabajar cómodamente en máquinas Unix remotas, con sesiones de texto
- También se puede ssh usar en Microsoft Windows, aunque con muchas limitaciones, resulta poco natural
- Pero habrá ocasiones en que será conveniente o imprescindible usar sesiones gráficas en máquinas remotas

Protocolos para sesiones gráficas remotas (1)

- Soluciones propietarias como LogMeIn o TeamViewer
- RDP. *Remote Desktop Services*, también conocido como *Terminal Services*.
  Nativo en Microsoft Windows, usable desde otras plataformas, p.e. `gnome-rdp`, `vinagre` (clientes) o `xrdp` (servidor)
  Puerto por omisión: 3389 TCP

Protocolos para sesiones gráficas remotas (2)

- X11 forwarding. Forma parte de X Window.
  Tradicional y nativo en Linux/UNIX, usable en Microsoft Windows.
  Normalmente no trabaja sobre un escritorio completo sino con ventanas individuales.
  Anticuado auque sigue disponible. Intrínsecamente poco seguro. Puerto por omisión: 6000 TCP
- VNC
  Protocolo abierto, multiplataforma. Nativo en muchos Linux. Disponible en Microsoft Windows, Unix, *BSD, macOS, Android, iOS, ...

X11 Forwarding

- X Window (año 1984) es el protocolo tradicional en Unix para mostrar gráficos. En local y también en remoto
  Nada que ver con Microsoft Windows
- En 1987 aparece la versión 11 de X Window, sigue siendo la versión actual. De ahí el nombre X11
- X Window es un protocolo cliente-servidor. Aunque la terminología puede ser anti-intuitiva:
  - La máquina local es el servidor. En ella están los gráficos. Normalmente lo llamaríamos *cliente*, pero es el *servidor X11* (ofrece el servicio de representación gráfica)
  - En la máquina remota está el *cliente X Window*. Aunque en esta máquina están los procesos que usan los gráficos, esto es, los servicios. (Los servicios son los clientes de los gráficos)

- En la máquina local puede ser necesario configurar los permisos del servidor
  `xhost +`
  permite que cualquier cliente X Window desde cualquier máquina lance ventanas en nuestro servidor X Window local.
  - También se pueden dar permisos más específicos
- Entre dos máquinas Ubuntu con el mismo usuario en ambas máquinas, mediante ssh, no es necesario dar permisos adicionales

Para lanzar una aplicación gráfica en una máquina remota:

- Desde la máquina local hacemos ssh a la máquina remota con la opción `-X` (mayúscula)
  `jperez@gamma12:~$ ssh -X alpha`
- Lanzamos la aplicación en segundo plano
  p.e
  `jperez@alpha:~$ xeyes&`

VNC

VNC, *Virtual Network Compute* es un protocolo para abrir sesiones gráficas en máquinas remotas

- Arquitectura cliente-servidor, desarrollado por The Olivetti & Oracle Research Lab
- La terminología es la habitual, no la de X Windows. El cliente está en la máquina local, el servidor, en la máquina remota
- Implementación liberada como software libre en 2002
- Muy popular. Muchas implementaciones para cualquier plataforma (Microsoft Windows, Linux, Unix, macOS, Android, iOS, Raspberry Pi ...)
  Cualquier servidor de cualquier plataforma puede trabajar con cualquier cliente en cualquier otra

En Ubuntu

- Tenemos un servidor integrado por omisión, llamado `vino`
- Hay un cliente llamado `vinagre`
- También podemos usar *TightVNC*, entre otras implementaciones

En Windows

- Podemos usar *TightVNC*, entre otras implementaciones

En macOS

- Como cliente podemos usar el navegador nativo de macOS, Safari. Basta escribir en la barra de direcciones la dirección del servidor:
  `vnc://maquina:puerto`
- Otra opción (con cliente y servidor) es RealVNC. Tiene una versión comercial y otra libre (RealVNC Open Edition)

En Raspbian (Raspberry Pi)

- El servidor VNC instalado es Real VNC, cuya configuración por omisión es incompatible con VNC estándar. Hay dos soluciones
  1. Usar el cliente de Real VNC (vncviewer), disponible en el web de Real VNC
  2. Configurar el servidor de Real VNC para que use autenticación VNC, no autenticación UNIX

vinagre

El cliente de VNC oficial de Ubuntu es `vinagre`

- Está desarrollado conjuntamente con `vino`, algunas opciones avanzadas pueden funcionar mejor con este servidor
- También tiene soporte para RDP

Uso:

```
vinagre <MAQUINA>:<PUERTO>
```

vino

- En Ubuntu, el servidor `vino` está instalado por omisión en las máquinas con escritorio gráfico
- Por omisión trabaja en el puerto 5900 TCP
- Para cambiar el puerto del servidor:
  1. Instalamos dconf-editor
     `sudo apt install dconf-editor`
  2. Lanzamos dconf-editor
  3. En
     `org | gnome | desktop | remote-access`
     Cambiamos `alternative-port`
     Activamos `use-alternative-port`

Normalmente usaremos vino cuando ya tenemos una sesión gráfica abierta en un Ubuntu con escritorio tradicional
Pero no es adecuado para abrir:

- Una segunda sesión gráfica en la misma máquina
- Una sesión gráfica en una máquina remota a la que no tenemos acceso físico o en la que no queremos abrir una sesión gráfica tradicional
- Una sesión en una máquina donde no queramos un escritorio tan pesado como Unity o Gnome

En cualquiera de estos casos

- En vez de usar *vino*, podremos usar *TightVNC*

Servidor de TightVNC en Ubuntu

El servidor de TightVNC es `vncserver`. Para usarlo necesitaremos, además del propio `vncserver`, un escritorio. Podríamos emplear Gnome o Unity, pero son muy pesados. Generalmente será más adecuado

- O bien un escritorio ligero como Xfce4
- O bien un gestor de ventanas como Openbox (podemos considerar a Openbox como un escritorio ultra-ligero)

Los pasos son:

1. Instalación de vncserver
2. Instalación del escritorio
3. Preparación del escritorio
4. Lanzamiento del servidor
5. Lanzamiento del cliente

1 Instalación de vncserver

En caso de que TightVNC no esté instalado en nuestro sistema

1. Como en cualquier instalación de un paquete nuevo, suele ser recomendable actualizar el sistema
   `sudo apt update; sudo apt upgrade`
2. Instalamos el paquete
   `sudo apt install tightvncserver`

2 Instalación del escritorio

Si vamos a usar Openbox

- Para saber si está instalado, intentamos ejecutar `openbox-session`
- Para instalarlo
  `sudo apt install openbox`

Si vamos a usar Xfce4

- Para saber si está instalado, intentamos ejecutar `xfce4-session`
- Para instalarlo
  `sudo apt install xfce4`

3 Preparación del escritorio

1. En el servidor escribiremos un fichero `~/.vnc/xstartup` con el siguiente contenido
   - Si vamos a usar Xfce4

     ```
     #!/bin/bash
     xrdb ~/.Xresources
     /usr/bin/xfce4-session
     ```

   - Si vamos a usar Openbox

     ```
     #!/bin/bash
     xrdb ~/.Xresources
     /usr/bin/openbox-session
     ```

2. Como a cualquier script, le damos permiso de ejecución, p.e. `chmod 755 ~/.vnc/xstartup`

4 Lanzamiento del servidor

Para lanzar el servidor ejecutamos la orden `vncserver`, indicando el tamaño de la pantalla y la profundidad del color, especificada en número de bits

Ejemplo:

```
vncserver  -geometry 1024x768 -depth 16
```

Si vamos a usarlo con frecuencia, puede ser conveniente poner esta orden en un script de shell, p.e.

```
~/bin/mi_vncserver
```

- La primera vez que lancemos `vncserver`, nos preguntará la contraseña de la sesión.
  Quedará almacenada en el fichero
  `~/.vnc/passwd`
- Si necesitamos cambiar la contraseña, ejecutamos
  `vncpasswd`
- También podemos cambiar la contraseña desde un script.
  P.e. la contraseña *sesamo* sería:
  `echo "sesamo\nsesamo\n\n" | vncpasswd`

Como cualquier demonio, `vncserver` deberá atarse a un puerto para aceptar peticiones, en su caso a un puerto TCP

- El puerto por omisión es el 5900 TCP
- Atención, el servidor de VNC no emplea el concepto *puerto TCP*, sino *display port*, donde
  puerto TCP = 5900 + *display port*
- Sin embargo, en el cliente normalmente sí indicaremos el puerto TCP, no el *display port*

Si no indicamos otra cosa, el demonio `vncserver` intentará usar el *display port* 0, puerto TCP 5900.

- Si está libre, mostrará el mensaje

  ```
  New 'X' desktop is gamma:0
  ```

- Si está ocupado (tal vez por vino), lo intentará en el *display port* 1, puerto TCP 5901. Si tiene éxito el mensaje será

  ```
  New 'X' desktop is gamma:1
  ```

  y así sucesivamente con los *display port* 2, 3, etc (puertos TCP 5902, TCP 5903, etc)

- También podemos indicar explícitamente el *display port* que usará el servidor:
  Ejemplo: *display port* 10 (puerto TCP 5910)
  `vncserver :10 -geometry 1024x768 -depth 16`
- La sesión permanecerá abierta hasta que la matemos explícitamente con
  `vncserver -kill :10`
  - Esta orden mata el proceso `vncserver`, la sesión X11 en `/tmp/.X11-unix` y los logs en `~/.vnc/`

5 Lanzamiento del cliente

En la máquina local ejecutamos
`vinagre <SERVIDOR>:<PUERTO`
p.e.
`vinagre gamma:5901`
Observa que en este caso indicamos el puerto TCP, no el *display port*

- Recuerda que para cerrar la sesión hay que matar el servidor, no basta con cerrar el cliente

Uso de VNC en Docker

VNC es una forma conveniente de abrir sesiones gráficas dentro de un contenedor.
Configuración de ejemplo para el *display port* 0, contraseña *sesamo* `entrypoint.sh`:

```
#!/bin/bash
vncserver :0 -geometry 1024x768 -depth 16
/usr/bin/xterm&
/bin/bash
```

Dockerfile:

```
FROM ubuntu:16.04
ENV USER root

RUN apt-get update && DEBIAN_FRONTEND=noninteractive &&  \
    apt-get install -y \
    tightvncserver openbox \
    xterm

# Expose VNC port
EXPOSE 5900

#set password for vnc
RUN echo "sesamo\nsesamo\n\n" | vncpasswd

COPY entrypoint.sh /
ENTRYPOINT ["/entrypoint.sh"]
```

http://ortuno.es/Dockerfile_vnc.txt

Lanzamiento del contendor:

```
#!/bin/bash
DISPLAY_NUMBER=0
PORT=$((DISPLAY_NUMBER+5900))
IMAGEN=vnc
NOMBRE=jper${IMAGEN}01
USUARIO=jperez

docker run -it --rm -h ${NOMBRE}  --name ${NOMBRE} -p ${PORT}:${PORT} \
    -e DISPLAY=:${DISPLAY_NUMBER} ${USUARIO}/${IMAGEN}
```

http://ortuno.es/lanza_vnc.txt