

Memoria práctica I

Grupo Viernes Mañana



16/02/2024

GUILLERMO BAJO LABORDA, 842748@unizar.es



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Resumen

Esta primera práctica de la asignatura de Administración de Sistemas 2 consiste en implementar un entorno de dos máquinas virtuales (VMs) utilizando OpenBSD sobre el entorno libvirt. Una de estas VMs funcionará como un router IPv6, mientras que la otra será una máquina de prueba para la red interna. Se desplegará un servicio de sincronización de tiempo (NTP) inicial y se desarrollará un script shell para administrar remotamente las VMs.

La práctica se divide en varias secciones que incluyen la configuración de imágenes base, la creación y configuración de un enrutador IPv6, el despliegue del servicio NTP y el desarrollo de un script de administración remota. A lo largo de la misma, se abordan otros aspectos como la configuración de interfaces de red, la generación de usuarios y la configuración de SSH para acceso remoto. Por otra parte, se utilizarán diversos servicios como el rad, que es un demonio de anuncio de red para IPv6. encargado de anunciar los prefijos de red y la información del encaminador por defecto a los nodos de la red, o el slaacd, un demonio de autoconfiguración de direcciones IPv6.

Finalmente, se desarrolla un script shell para gestionar las VMs de forma remota, permitiendo operaciones como definir, iniciar y detener las VMs, con soporte para gestionar grupos de VMs a través de un archivo de configuración.

Introducción y objetivos

El objetivo principal de esta primera práctica de la asignatura de Administración de Sistemas 2 es llevar a cabo el despliegue de dos máquinas virtuales (VMs) utilizando el sistema operativo OpenBSD en un entorno de virtualización que emplea libvirt como herramienta principal. Estas VMs serán configuradas para operar dentro de una red virtualizada, donde una de ellas será designada como router y se configurará específicamente para gestionar el tráfico IPv6 (será responsable de dirigir el tráfico entre las otras máquinas conectadas a la red, utilizando el protocolo IPv6 para el enrutamiento).

Además de la configuración de red, se implementará un servicio de sincronización de tiempo inicial utilizando el protocolo NTP (Network Time Protocol). Esto garantizará una sincronización precisa del tiempo entre las VMs y con los servidores de tiempo externos, lo que es crucial para mantener la coherencia en las operaciones de red y los registros de eventos.

Para facilitar la administración remota de las VMs, se desarrollarán programas shell específicos que permitirán llevar a cabo tareas de gestión y supervisión de manera eficiente y desde cualquier ubicación. Estos scripts proporcionarán funcionalidades como el inicio y detención de las VMs, la monitorización del estado de la red y la gestión de la configuración del router IPv6, entre otras posibles funcionalidades según los requisitos específicos del entorno virtualizado.

Arquitectura de elementos relevantes

Configuración de la imagen base

Primeramente, se han generado un par de claves localmente y se han copiado a central y de central a lab para poder conectarse mediante SSH sin contraseña. También se ha configurado el acceso mediante SSH sin contraseña entre las máquinas del laboratorio y la máquina virtual o1B, definida en esta sección B.

Creación y configuración básica de un encaminador IPv6

Primeramente se ha creado una imagen diferencial orouter1B.qcow2, la cual crea un fichero donde sólo guardar las diferencias mientras utiliza los datos no cambiados del fichero de imagen base. Se define la máquina correspondiente en libvirt y se arranca la VM conectándonos como root.

Se modifica el contenido de /etc/hostname.vio0 para que funcione solo con la dirección que especificamos a continuación:

```
Unset
inet6 alias 2001:470:736b:f000:11b1
-autoconfprivacy
up
```

La primera línea indica que se debe asignar una dirección IPv6 adicional a la interfaz vio0. La segunda es una opción que deshabilita la generación automática de direcciones IPv6 privadas temporales para esta interfaz. Al desactivar esta función, la dirección IPv6 configurada se utilizará de manera estática y no cambiará automáticamente.

También modificamos el fichero /etc/mygate:

```
Unset
2001:470:736b:f000::1
```

Al incluir solo el encaminador IPv6 principal por defecto de la subred (router central del esquema del enunciado de la práctica), estamos configurando la dirección del router principal de la red para que se utilice como la puerta de enlace predeterminada para cualquier tráfico destinado a redes fuera de la subred local.

A continuación, creamos una nueva configuración de red para la interfaz virtual `vlan2799`, que está asociada con la VLAN 2799, utilizando la dirección IPv6 `2001:470:736b:1bff::1`. Esta interfaz virtual estará conectada a través de la interfaz física `vio0`, y esta configuración nos permitirá establecer la conectividad en una red específica. Para ello, se ha establecido el siguiente contenido en el fichero `/etc/hostname.vlan2799`:

```
Unset
inet6 alias 2001:470:736b:1bff::1 64 vlan 2799 vlandev vio0
-autoconfprivacy
```

Con el fin de activar encaminamiento ip6 y no contestación de anuncios de prefijo ip6, se ha modificado `/etc/sysctl.conf` con `net.inet6.ip6.forwarding=1`. A su vez, para poner en funcionamiento el servicio de anuncio de prefijos IPv6 a la subred de la vlan, se ha añadido al fichero `/etc/rc.conf.local` la línea `rad_flags=""`, y se ha creado el fichero `/etc/rad.conf` con la línea `interface vlan 2799`.

Creación de otra VM de prueba para la red interior

Se ha creado otra imagen diferencial de forma bastante similar, realizando las respectivas modificaciones en el fichero xml.

Modificamos el fichero `/etc/hostname.vio0` de la nueva máquina virtual y añadimos la siguiente configuración:

```
Unset
-inet6
inet6 -soii
inet6 -temporary
-autoconfprivacy
```

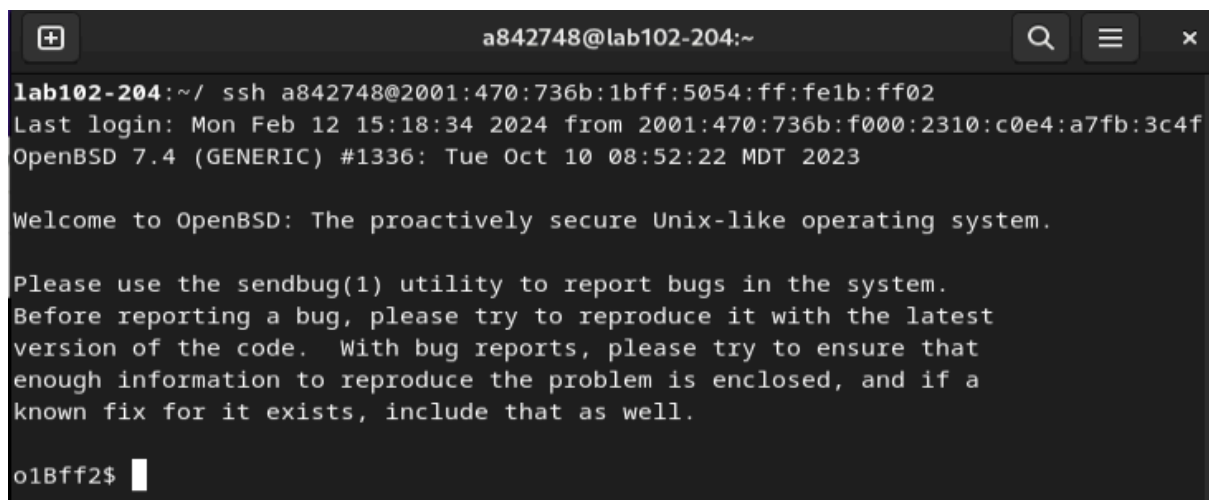
La primera línea indica que se debe desactivar la configuración automática de IPv6 en la interfaz `vio0`. `Up` indica que la interfaz de red `vio0` debe estar activa. Luego se indica que se debe desactivar la autoconfiguración de direcciones de origen y desactivar la autoconfiguración de direcciones de interfaz. Finalmente, la última indica que se deben desactivar las direcciones IPv6 temporales en la interfaz `vio0`.

De nuevo, creamos el `/etc/hostname.vlan2799`:

```
Unset
vlan 2799 vlandev vio0 up
inet6 autoconf
inet6 -soii
inet6 -temporary
-autoconfprivacy
```

Es necesario habilitar el servicio *slaacd* para permitir que el sistema configure automáticamente sus direcciones IPv6 y participe en la red IPv6 sin necesidad de una configuración manual. Lo habilitamos con el comando *doas rcctl enable slaacd*.

Tras realizar esta configuración, podemos acceder a la máquina o1Bff2 a través del encaminador orouter1B:



```
a842748@lab102-204:~
lab102-204:~/ ssh a842748@2001:470:736b:1bff:5054:ff:fe1b:ff02
Last login: Mon Feb 12 15:18:34 2024 from 2001:470:736b:f000:2310:c0e4:a7fb:3c4f
OpenBSD 7.4 (GENERIC) #1336: Tue Oct 10 08:52:22 MDT 2023

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

o1Bff2$
```

[Servicio de sincronización de tiempo](#)

Para poner en marcha el servicio de sincronización de tiempo cuyo servidor ntp es el encaminador configurado en el apartado C, se han realizado ciertas modificaciones en los siguientes ficheros:

En la máquina orouter1B, en el fichero `/etc/ntp.conf` añadimos las siguientes líneas:

```
Unset
server 2001:470:0:50::2
server 2001:470:0:2c8::2
listen on 2001:470:736b:1bff::1
```

Y en la máquina o1Bff2, en el fichero /etc/ntp.conf nuevamente añadimos:

```
Unset
server ntp1
```

En el /etc/hosts de esta última máquina añadimos la siguiente sentencia, lo que significa que cuando se intenta acceder a ntp1, el sistema buscará esta dirección IPv6 en el fichero /etc/hosts.

```
Unset
2001:470:736b:1bff:1 ntp o1Bff2
```

Desarrollo de script shell para gestión remota de VMs con libvirt (virsh).

Finalmente, se ha elaborado un script en shell para facilitar la gestión remota de estas máquinas virtuales. Es un script bastante sencillo que tiene 2 usos: `./pr1_adsis.sh [fichero]`, donde se puede especificar un fichero de configuración o no. En este fichero de configuración se pueden especificar un listado de máquinas y lo que se desea hacer con cada una (ponerlas en marcha, detenerlas, o incluso definir nuevas). Si no se especifica un fichero, el usuario simplemente deberá introducir manualmente el nombre de las máquinas y la opción deseada, hasta que haya acabado y escoja la opción de salir.

El Script detallado se muestra a continuación:

```
Unset
#!/bin/bash

# Este script permite definir, poner en marcha o detener una VM en un servidor
remoto.
# Tiene dos modos de uso:
# 1. Interactivo: Si no se proporciona ningún parámetro al ejecutar el script, se
mostrarán opciones al usuario para interactuar con el script.
# 2. Desde un archivo de configuración: Si se proporciona un archivo de
configuración como parámetro al ejecutar el script, leerá las instrucciones del
archivo y ejecutará las acciones correspondientes.

# Verificar el número de parámetros proporcionados
if [ $# -eq 0 ]; then
    # Mostrar opciones al usuario
    echo "Este script le permitirá definir, poner en marcha o detener una VM hasta que
escoja la opción de salir."
    while true; do
```

```

echo "Opciones:"
echo "1. Definir una VM"
echo "2. Poner en marcha una VM"
echo "3. Detener una VM"
echo "4. Salir"

# Leer la opción del usuario
read opcion

# Comprobar la opción seleccionada y ejecutar la acción correspondiente
case $opcion in
    1)
        echo "Introduzca el nombre de la VM que desea definir:"
        read nombre
        ssh -n a842748@155.210.154.204 "cd
/../../../../misc/alumnos/as2/as22023/a842748 && virsh -c qemu:///system define
$nombre.xml"
        ;;
    2)
        echo "Introduzca el nombre de la VM que desea poner en marcha:"
        read nombre
        ssh a842748@155.210.154.204 "virsh -c qemu:///system start $nombre"
        ;;
    3)
        echo "Introduzca el nombre de la VM que desea detener:"
        read nombre
        ssh a842748@155.210.154.204 "virsh -c qemu:///system shutdown $nombre"
        ;;
    4)
        echo "Ha seleccionado salir, ¡adiós!"
        exit 0
        ;;
    *)
        echo "Error: Opción no válida. Por favor, seleccione una opción del 1 al 4."
        exit 1
        ;;
esac
done

elif [ $# -eq 1 ]; then
    archivo=$1
    if [ ! -f $archivo ]; then
        echo "El archivo $archivo no existe."
        exit 1
    fi

    while IFS=' ' read -r codigo nombre_vm; do
        case $codigo in

```



```
1)
    ssh -n a842748@155.210.154.204 "cd ~/misc/alumnos/as2/as22023/a842748 &&
virsh -c qemu:///system define $nombre_vm"
    ;;
2)
    ssh -n a842748@155.210.154.204 "virsh -c qemu:///system start $nombre_vm"
    ;;
3)
    ssh -n a842748@155.210.154.204 "virsh -c qemu:///system shutdown
$nombre_vm"
    ;;
*)
    echo "Error: Número de acción no válido en el archivo."
    exit 1
    ;;
esac
done < $archivo
else
    echo "Numero de parametros incorrecto. Usage: ./pr1_adsis.sh [fichero_config]"
fi
```

Comprensión de elementos significativos de la práctica

¿Qué ocurre si introducimos "inet6 autoconf" en el fichero "hostname.vio0" de la máquina interna de prueba y por qué?

Si introducimos "inet6 autoconf" en el archivo "hostname.vio0" de la máquina interna de prueba, lo que estamos haciendo es habilitar la autoconfiguración de direcciones IPv6 para la interfaz de red vio0. Esto significa que la interfaz intentará obtener automáticamente una dirección IPv6 única y válida utilizando el mecanismo de autoconfiguración de direcciones IPv6

Problemas encontrados y su solución

El principal problema encontrado en esta práctica ha sido nada más empezar, cuando al intentar añadir una conexión en Virtual Machine Manager con cualquiera de las máquinas del laboratorio que estuvieran encendidas, daba errores en la configuración de SSH. Se buscaron distintas alternativas frente a este problema, de las cuales ninguna funcionó. Se volvió a intentar añadir la conexión a las horas y finalmente funcionó, sin haber hecho nada distinto a lo hecho anteriormente.

Por otra parte, en el paso D, tras hacer la configuración correspondiente y consultar las interfaces de red con `ifconfig`, en la interfaz `vlan2799` solamente me aparecía la dirección local link. Tras revisar los distintos ficheros modificados a lo largo de esta práctica, me percaté de que en `/etc/rad.conf` de `orouter1B`, especifiqué mal la interfaz poniendo `vio0`, en vez de `vlan2799`. Tras realizar la correspondiente modificación, ya me mostraba las direcciones correctamente al hacer `ifconfig` desde `o1Bff2`.

También hubo problemas con la configuración de la red VLAN, se podía hacer SSH a la interfaz `vio0` del router pero no al VLAN. Finalmente, el error residía en que en vez de copiar el fichero `/etc/hostname.vio0` a el fichero `/etc/hostname.vlan2799`, creé este segundo de cero. Esto puede deberse a que al crear el fichero de cero, este no tuviera los mismos permisos. Tras eliminar el fichero y copiarlo como especificaba el enunciado, ya funcionó la conexión SSH desde CentOs al router y a `o1Bff2`.

La última situación problemática encontrada, fue causada por apagar la máquina del laboratorio sin haber hecho antes `shutdown` al router, lo que causó que al iniciar el router de nuevo este no funcionase y mostrase errores de Core Dump. Por suerte, realicé una copia de seguridad del router pocos días antes y tenía una versión no actualizada del todo, pero prácticamente igual, por lo que solo hizo falta restaurarla y realizar cambios puntuales para recuperar su correcto funcionamiento.

En conclusión, la mayoría de problemas fueron causadas por situaciones puntuales y meros errores propios.