Centro de Enseñanza Técnica Industrial

Desarrollo de Software

Proyecto de segundo parcial

Jesús Alberto Aréchiga Carrillo

22310439 5N

Profesor

José Francisco Pérez Reyes

Octubre 2024

Guadalajara, Jalisco

Introducción

Inicialmente se tienen las VMs están en modo bridge, se tiene que cambiar la configuración para que en vez de que sea modo bridge, sea NAT. Una vez que se hagan todas las modificaciones, se necesita configurar el Dom0 para que pueda enmascarar paquetes y los pueda redireccionar a una red privada que es donde están las máquinas virtuales.

Desarrollo

Eliminar interfaz y configuraciones de bridge (si aplica)

El primer cambio por el que se puede empezar será poner el Dom0 en modo nat y quitar el puente que se le había hecho anteriormente con los comandos “ip link set xenbr0 down” y el comando “ip link delete xenbr0”. Esto suponiendo que xenbr0 es la interfaz virtual que se había configurado anteriormente.

```
root@a22310439:/# ip link set xenbr0 down
```

```
root@a22310439:/# ip link delete xenbr0
```

También se cambia la configuración de modo puente que se tuvo que configurar desde un inicio, en el archivo /etc/xen/xl.conf solo se tiene que cambiar la línea que dice vif.default.script=”vif-bridge” a que sea “vif-nat”.

```
# default option to run hotplug scripts from xl
# if disabled the old behaviour will be used, and hotplug scripts will be
# launched by udev.
#run_hotplug_scripts=1

# default backend domain to connect guest vifs to.  This can be any
# valid domain identifier.
#vif.default.backend="0"

# default gateway device to use with vif-route hotplug script
#vif.default.gatewaydev="eth0"

# default vif script to use if none is specified in the guest config
vif.default.script="vif-nat"

# default bridge device to use with vif-bridge hotplug scripts
#vif.default.bridge="xenbr0"

# Reserve a claim of memory when launching a guest. This guarantees immediate
# feedback whether the guest can be launched due to memory exhaustion
# (which can take a long time to find out if launching huge guests).
# see xl.conf(5) for details.
#claim_mode=1

# Specify global vcpu hard affinity masks. See xl.conf(5) for details.
```

Se habilita la redirección de IPv4, esto se puede hacer de 2 maneras, con el comando “sysctl net.ipv4.ip_forward=1” o modificando el archivo /etc/sysctl.conf y habilitando esa línea en el archivo.

```
#kernel.printk = 3 4 1 3

##############################################################
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```

Configuración de las tablas de IP (iptables)

Se configuran las tablas de IPs del Dom0, primero se tiene que instalar el paquete iptables con el comando “apt-get install iptables”

Luego, se tiene que habilitar el tráfico que proviene de la red creada con anterioridad a través de la interfaz física, que es la que si tiene salida a internet.

- iptables -t nat -A POSTROUTING -s 10.0.0.024 -o enp2s0 -j MASQUERADE
- iptables -A FORWARD -m conntrack -- ctstate RELATED, ESTABLISHED -j ACCEPT
- iptables -A FORWARD -s 10.0.0.0/24 -o enp2s0 -j ACCEPT

(Tómese en cuenta que aquí se utiliza enp2s0 porque es la interfaz física que se está utilizando, este campo se cambia dependiendo de la interfaz con la que se cuente. Para saber cual interfaz se está utilizando, se utiliza el comando “ip a s”)

Después se configura el paso del DNS al servidor DNS de la red. Si no se tiene un servidor DNS en la red, se puede configurar uno público, en este caso se va a utilizar el de Google 8.8.8.8

- iptables -t nat -A PREROUTING -s 10.0.0.0/24 -p udp --dport 53 -j DNAT --to - destination 8.8.8.8
- iptables -t nat -A PREROUTING -s 10.0.0.0/24 -p tcp --dport 53 -j DNAT --to - destination 8.8.8.8

Se revisa que las tablas se aplicaron correctamente con el comando “iptables-save”

```
root@a22310439:/# iptables-save
# Generated by iptables-save v1.8.9 (nf_tables) on Tue Oct 29 03:54:47 2024
*filter
:INPUT ACCEPT [47625:7825398]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [4165:460906]
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 10.0.0.0/24 -o enp2s0 -j ACCEPT
COMMIT
# Completed on Tue Oct 29 03:54:47 2024
# Generated by iptables-save v1.8.9 (nf_tables) on Tue Oct 29 03:54:47 2024
*nat
:PREROUTING ACCEPT [724:68725]
:INPUT ACCEPT [713:66885]
:OUTPUT ACCEPT [92:6906]
:POSTROUTING ACCEPT [92:6906]
-A PREROUTING -s 10.0.0.0/24 -p udp -m udp --dport 53 -j DNAT --to-destination 8.8.8.8
-A PREROUTING -s 10.0.0.0/24 -p tcp -m tcp --dport 53 -j DNAT --to-destination 8.8.8.8
-A POSTROUTING -s 10.0.0.0/24 -o enp2s0 -j MASQUERADE
COMMIT
# Completed on Tue Oct 29 03:54:47 2024
```

Como se puede ver, todas las configuraciones ingresadas anteriormente están presentes en la salida del comando ejecutado.

Una vez que se corrobora que las tablas de IP están correctas, se puede reiniciar el servicio de red con el comando “systemctl restart networking” o “/etc/init.d/networking restart”.

Es importante notar que las configuraciones de las tablas IP no son permanentes, es decir, que al reiniciar el Dom0, las configuraciones se perderán, por lo que hay que volver a ejecutar los comandos de iptables anteriores cuando se inicie el sistema. Se pueden guardar en un archivo bash y configurar para que se ejecute al bootear.

Configuración de IPs de las máquinas virtuales

Ahora se tienen que configurar las IPs de las VMs, estas van a ser estáticas y para que puedan tener comunicación entre ellas, deben estar en el mismo segmento de red (penúltimo octeto). En este caso se va a estar usando la red 10.0.0.0/24

Primero asegurarnos que no hay ninguna VM en ejecución con el comando “xl list” o “xentop”.

Es importante tener en cuenta que la IP que tenga la VM se conectará al Dom0 con una IP incrementada en 127, por ende, la IP de la VM tiene que ser menor a 128.

Las IPs que se van a usar son:

- Pruebaxen: 10.0.0.3
- Vsftpd: 10.0.0.4
- WebDAV: 10.0.0.5
- http: 10.0.0.6
- DNS: 10.0.0.7

Para esto, necesitamos montar el disco de cada VM y cambiar la configuración de red. Para montar el disco de una VM primero se tiene que crear una carpeta temporal que será donde se monte el disco.

```
root@a22310439:~# mkdir temp
root@a22310439:~# ls -la
total 48
drwx------  6 root root 4096 Oct 29 04:10 .
drwxr-xr-x 19 root root 4096 Oct 29 03:54 ..
-rw-------  1 root root 2764 Oct 26 05:31 .bash_history
-rw-r--r--  1 root root  571 Apr 10  2021 .bashrc
drwx------  2 root root 4096 Oct 23 01:07 .cache
drwxr-xr-x  3 root root 4096 Oct 26 05:18 .openjfx
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
drwx------  2 root root 4096 Oct 23 00:52 .ssh
drwxr-xr-x  2 root root 4096 Oct 29 04:10 temp
-rw-------  1 root root 9687 Oct 29 03:54 .viminfo
```

Ahora se puede montar el disco de la VM en la carpeta temporal.

```
root@a22310439:~# ls -la /temp
```

```
root@a22310439:~# ls -la temp
total 84
drwxr-xr-x 18 root root  4096 Oct 23 02:13 .
drwx------  6 root root  4096 Oct 29 04:10 ..
lrwxrwxrwx  1 root root     7 Oct 23 02:11 bin -> usr/bin
drwxr-xr-x  3 root root  4096 Oct 23 02:13 boot
drwxr-xr-x  4 root root  4096 Oct 23 02:11 dev
drwxr-xr-x 52 root root  4096 Oct 29 01:25 etc
drwxr-xr-x  2 root root  4096 Aug 14 11:10 home
lrwxrwxrwx  1 root root    30 Oct 23 02:13 initrd.img -> boot/initrd.img-6.1.0-26-amd64
lrwxrwxrwx  1 root root    30 Oct 23 02:13 initrd.img.old -> boot/initrd.img-6.1.0-26-amd64
lrwxrwxrwx  1 root root     7 Oct 23 02:11 lib -> usr/lib
lrwxrwxrwx  1 root root     9 Oct 23 02:13 lib32 -> usr/lib32
lrwxrwxrwx  1 root root     9 Oct 23 02:11 lib64 -> usr/lib64
drwx------  2 root root 16384 Oct 23 02:10 lost+found
drwxr-xr-x  2 root root  4096 Oct 23 02:11 media
drwxr-xr-x  2 root root  4096 Oct 23 02:11 mnt
drwxr-xr-x  2 root root  4096 Oct 23 02:11 opt
drwxr-xr-x  2 root root  4096 Aug 14 11:10 proc
drwx------  4 root root  4096 Oct 29 04:06 root
drwxr-xr-x  9 root root  4096 Oct 23 02:12 run
lrwxrwxrwx  1 root root     8 Oct 23 02:11 sbin -> usr/sbin
drwxr-xr-x  2 root root  4096 Oct 23 02:11 srv
drwxr-xr-x  2 root root  4096 Aug 14 11:10 sys
drwxrwxrwt  6 root root  4096 Oct 29 03:56 tmp
drwxr-xr-x 13 root root  4096 Oct 23 02:13 usr
drwxr-xr-x 11 root root  4096 Oct 23 02:11 var
lrwxrwxrwx  1 root root    27 Oct 23 02:13 vmlinuz -> boot/vmlinuz-6.1.0-26-amd64
lrwxrwxrwx  1 root root    27 Oct 23 02:13 vmlinuz.old -> boot/vmlinuz-6.1.0-26-amd64
```

Se puede ver que todos los archivos de la VM están en la carpeta temp.

```
root@a22310439:~# vim temp/etc/network/interfaces
```

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enX0
iface enX0 inet static
        address 10.0.0.3/24
        gateway 10.0.0.130
        dns-nameservers 10.0.0.130


 # post-up  ethtool -K eth0 tx off
```

Una vez hecho el cambio se puede salir de la carpeta y desmontar el disco.

```
root@a22310439:~# umount temp
```

Tip: Se recomienda cambiar la configuración de la VM para que inicie directamente con la IP definida. Esto se puede hacer desde el archivo /etc/xen/pruebaxen.cfg (el archivo de configuración de la VM)

```
root@a22310439:~# vim /etc/xen/pruebaxen.cfg
```

```
#
#  Networking
#
vif         = [ 'mac=00:16:3E:00:70:35, ip=10.0.0.3' ]
```

Se tiene que dejar la dirección MAC y la IP definida para que haya consistencia con la configuración de red de la VM.

Se tiene que repetir el proceso para cada VM que se quiera levantar.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enX0
iface enX0 inet static
        address 10.0.0.4/24
        gateway 10.0.0.130
        dns-nameservers 10.0.0.130


 # post-up  ethtool -K eth0 tx off
```

Configuración de vsftpd

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enX0
iface enX0 inet static
        address 10.0.0.5/24
        gateway 10.0.0.130
        dns-nameservers 10.0.0.130
```

Configuración de http

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enX0
iface enX0 inet static
        address 10.0.0.6/24
        gateway 10.0.0.130
        dns-nameservers 10.0.0.130
```

Configuración de webdav

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enX0
iface enX0 inet static
        address 10.0.0.7/24
        gateway 10.0.0.130
        dns-nameservers 10.0.0.130
```

Configuración de dns

Se puede corroborar la salida a internet desde haciendo ping a algún servidor o con el comando “apt update”

```
root@dns:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=10.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=11.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=11.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=10.5 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=118 time=10.0 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 10.014/10.599/11.062/0.395 ms
root@dns:~# apt update
Hit:1 http://security.debian.org bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
N: Repository 'Debian bookworm' changed its 'non-free component' value from 'non-free' to 'non-free non-free-f
irmware'
N: More information about this can be found online in the Release notes at: https://www.debian.org/releases/bo
okworm/amd64/release-notes/ch-information.html#non-free-split
```

Ya que vimos que, si tiene internet la VM, podemos verificar las IPs que fueron asignadas en el Dom0

```
root@a22310439:~# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
    link/ether b0:6e:bf:4e:4c:cf brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.88/24 brd 192.168.1.255 scope global dynamic noprefixroute enp2s0
       valid_lft 67547sec preferred_lft 67547sec
    inet6 2806:102e:1e:1c95:a123:d2ec:f592:4026/64 scope global temporary dynamic
       valid_lft 86150sec preferred_lft 67275sec
    inet6 2806:102e:1e:1c95:b26e:bfff:fe4e:4ccf/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86150sec preferred_lft 86150sec
    inet6 fe80::b26e:bfff:fe4e:4ccf/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlp4s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 1a:bc:5e:d7:46:88 brd ff:ff:ff:ff:ff:ff permaddr b0:35:9f:bd:d1:1b
4: vif1.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.130/32 scope global vif1.0
       valid_lft forever preferred_lft forever
5: vif2.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.131/32 scope global vif2.0
       valid_lft forever preferred_lft forever
6: vif3.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.132/32 scope global vif3.0
       valid_lft forever preferred_lft forever
7: vif4.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.133/32 scope global vif4.0
       valid_lft forever preferred_lft forever
8: vif5.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.134/32 scope global vif5.0
```

Desde el Dom0 se ejecuta el comando “ip a s” y se puede verificar que las interfaces 4, 5, 6, 7 y 8 son interfaces creadas con los números de las VMs y tienen la IP que se les asignaron anteriormente aumentadas en 127.