

Práctica 3

Jesús Rodríguez Heras
Arantzazu Ota Alberro

4 de abril de 2019

1. Instalación de máquinas virtuales mediante Vagrant

En esta primera parte vamos a crear el entorno de trabajo, consiste en dos redes internas, conectadas al exterior mediante un router.

- La primera red tendrá el rango de IPs 192.168.2.0.
- La segunda red tendrá el rango de IPs 192.168.3.0.

Cada red tendrá un par de máquinas virtuales (no hace falta conectarlas todas de forma simultánea). Además, las redes solo tendrán acceso al exterior a través de la máquina que actúa como router.

En este ejercicio se deberá:

- **Crear el entorno de red mediante un único fichero Vagrant.**

```
1 Vagrant.configure("2") do |config|
2   config.vm.box = "debian/jessie64"
3
4   config.vm.define :r do |r|
5     r.vm.box="debian/jessie64"
6     r.vm.hostname="R"
7     r.vm.network "private_network", ip: "192.168.2.1"
8     r.vm.network "private_network", ip: "192.168.3.1"
9   end
10
11  config.vm.define :vm1r1 do |vm1r1|
12    vm1r1.vm.box="debian/jessie64"
13    vm1r1.vm.hostname="VM1"
14    vm1r1.vm.network "private_network", ip: "192.168.2.2"
15  end
16
17  config.vm.define :vm2r1 do |vm2r1|
18    vm2r1.vm.box="debian/jessie64"
19    vm2r1.vm.hostname="VM2"
20    vm2r1.vm.network "private_network", ip: "192.168.2.3"
21  end
22
23  config.vm.define :vm1r2 do |vm1r2|
24    vm1r2.vm.box="debian/jessie64"
25    vm1r2.vm.hostname="VM3"
26    vm1r2.vm.network "private_network", ip: "192.168.3.2"
27  end
28
29  config.vm.define :vm2r2 do |vm2r2|
30    vm2r2.vm.box="debian/jessie64"
31    vm2r2.vm.hostname="VM4"
32    vm2r2.vm.network "private_network", ip: "192.168.3.3"
33  end
34 end
```

■ Configurar el cortafuegos para que de acceso al exterior.

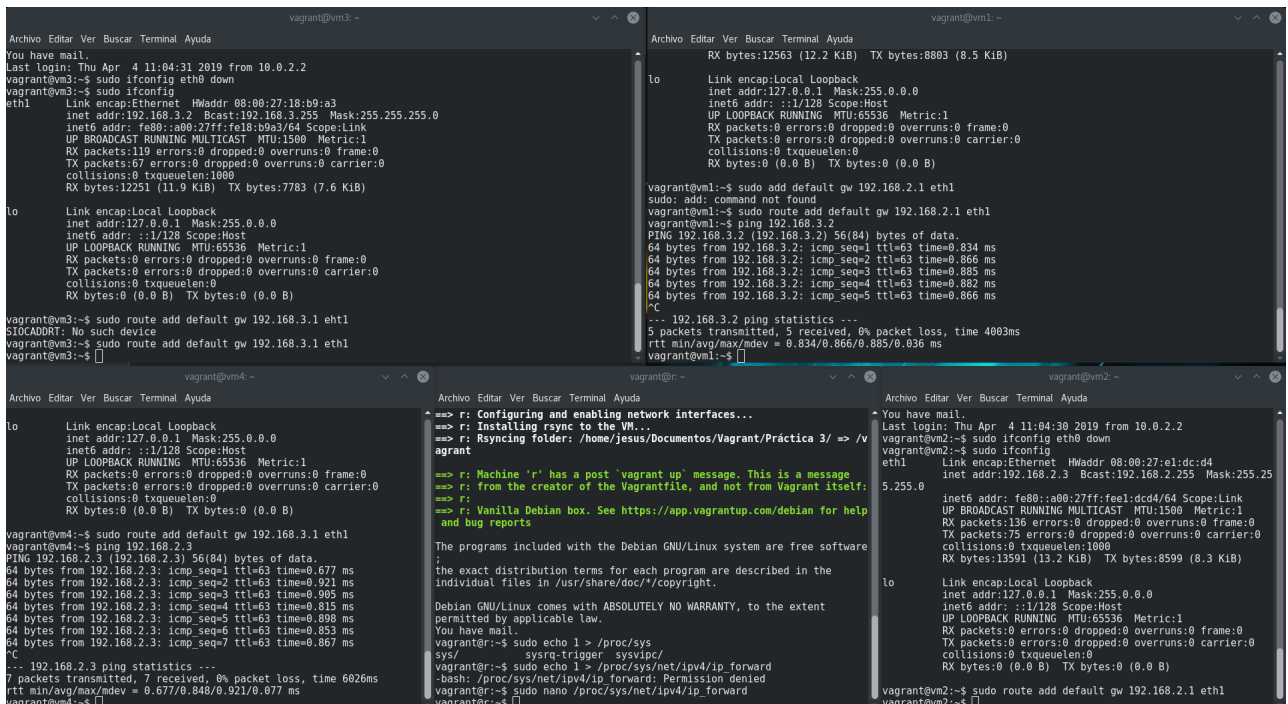
Para configurar el cortafuegos, primero debemos deshabilitar la interfaz de red que nos permite salir a Internet desde cada una de las máquinas de las redes. Para ello, identificamos la interfaz que queremos deshabilitar (en nuestro caso, `eth0`).

Si deshabilitamos esta interfaz directamente, tendremos el problema de que la terminal se quedará colgada porque estamos conectados por SSH a esa interfaz, entonces debemos habilitar la conexión SSH desde la otra interfaz. Para ello nos dirigimos al fichero `sshd_config` (`sudo nano /etc/ssh/sshd_config`) y descomentamos la línea que dice `PasswordAuthentication yes`. Luego, reiniciamos el servicio SSH (`sudo /etc/init.d/ssh restart`) y ya podemos conectarnos desde la máquina que actúa como router/cortafuegos.

Es hora de entrar en la máquina cortafuegos y conectarnos por SSH a cada una de las máquinas. Una vez dentro de las máquinas servidoras usamos el comando `sudo ifconfig eth0 down` para deshabilitar la interfaz de red externa.

También debemos activar el ip forward en el router para tener conectividad entre las máquinas de las diferentes redes. Para ello usamos el comando `sudo echo 1 >/proc/sys/net/ipv4/ip_forward`.

A continuación, establecemos la puerta de enlace como la interfaz que tenemos en la máquina que hace de router. Para ello usamos el comando `sudo route add default gw 192.168.x.1 eth1`, siendo `x` la red a la que pertenece cada máquina, y `eth1` la interfaz de la red interna.



```

vagrant@vm3: ~
$ sudo ifconfig eth0 down
$ sudo ifconfig
eth1    Link encap:Ethernet  HWaddr 08:00:27:18:b0:a3
        inet addr:192.168.3.2  Bcast:192.168.3.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe18:b9a3/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:119 errors:0 dropped:0 overruns:0 frame:0
        TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:12251 (11.9 KiB)  TX bytes:7783 (7.6 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vagrant@vm3:~$ sudo route add default gw 192.168.3.1 eth1
SIOCADDRT: No such device
vagrant@vm3:~$ sudo route add default gw 192.168.3.1 eth1
vagrant@vm3:~$

vagrant@vm4: ~
$ sudo route add default gw 192.168.3.1 eth1
$ ping 192.168.2.3
PING 192.168.2.3 (192.168.2.3) 56(84) bytes of data:
64 bytes from 192.168.2.3: icmp_seq=1 ttl=63 time=0.677 ms
64 bytes from 192.168.2.3: icmp_seq=2 ttl=63 time=0.921 ms
64 bytes from 192.168.2.3: icmp_seq=3 ttl=63 time=0.905 ms
64 bytes from 192.168.2.3: icmp_seq=4 ttl=63 time=0.815 ms
64 bytes from 192.168.2.3: icmp_seq=5 ttl=63 time=0.898 ms
64 bytes from 192.168.2.3: icmp_seq=6 ttl=63 time=0.853 ms
64 bytes from 192.168.2.3: icmp_seq=7 ttl=63 time=0.867 ms
^C
--- 192.168.2.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6026ms
rtt min/avg/max/mdev = 0.677/0.848/0.921/0.077 ms
vagrant@vm4:~$

vagrant@vm2: ~
$ sudo ifconfig eth0 down
$ sudo ifconfig
eth1    Link encap:Ethernet  HWaddr 08:00:27:e1:dc:d4
        inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.0.0
        inet6 addr: fe80::a00:27ff:fe1:dc:d4/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:136 errors:0 dropped:0 overruns:0 frame:0
        TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:13591 (13.2 KiB)  TX bytes:8599 (8.3 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vagrant@vm2:~$ sudo route add default gw 192.168.2.1 eth1
vagrant@vm2:~$

vagrant@vm1: ~
$ sudo add default gw 192.168.2.1 eth1
sudo: add: command not found
vagrant@vm1:~$ sudo route add default gw 192.168.2.1 eth1
vagrant@vm1:~$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data:
64 bytes from 192.168.3.2: icmp_seq=1 ttl=63 time=0.834 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=63 time=0.866 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=63 time=0.885 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=63 time=0.882 ms
64 bytes from 192.168.3.2: icmp_seq=5 ttl=63 time=0.866 ms
^C
--- 192.168.3.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.834/0.866/0.885/0.036 ms
vagrant@vm1:~$
```

Figura 1: En esta imagen se ve como la vm4 (red 2) hace ping a la vm2 (red 1) y la vm1 (red1) hace ping a la vm3 (red 2).

Lo siguiente es habilitar el enrutamiento a partir de la máquina router al resto de máquinas. El estado inicial es el siguiente:

```
vagrant@vm3: ~$ sudo ifconfig eth0 down
vagrant@vm3: ~$ sudo ifconfig eth1
Link encap:Ethernet  HWaddr 08:00:27:18:b9:a3
inet addr:192.168.3.2  Bcast:192.168.3.255  Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe18:b9a3/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:119 errors:0 dropped:0 overruns:0 frame:0
TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:12251 (11.9 KiB)  TX bytes:7783 (7.6 KiB)

lo
Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING  MTU:65536  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vagrant@vm3:~$ sudo route add default gw 192.168.3.1 eth1
SIOCADORT: No such device
vagrant@vm3:~$ sudo route add default gw 192.168.3.1 eth1
vagrant@vm3:~$ ping google.es
ping: unknown host google.es
vagrant@vm3:~$
```

```
vagrant@vm1: ~$ sudo add default gw 192.168.2.1 eth1
sudo: add: command not found
vagrant@vm1:~$ sudo route add default gw 192.168.2.1 eth1
vagrant@vm1:~$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data:
64 bytes from 192.168.3.2: icmp_seq=1 ttl=63 time=0.834 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=63 time=0.866 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=63 time=0.885 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=63 time=0.882 ms
64 bytes from 192.168.3.2: icmp_seq=5 ttl=63 time=0.866 ms
^C
--- 192.168.3.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/ndev = 0.834/0.866/0.885/0.036 ms
vagrant@vm1:~$ ping google.es
ping: unknown host google.es
vagrant@vm1:~$
```

```
vagrant@vm4: ~$ sudo route add default gw 192.168.3.1 eth1
PING 192.168.2.3 (192.168.2.3) 56(84) bytes of data:
64 bytes from 192.168.2.3: icmp_seq=1 ttl=63 time=0.677 ms
64 bytes from 192.168.2.3: icmp_seq=2 ttl=63 time=0.921 ms
64 bytes from 192.168.2.3: icmp_seq=3 ttl=63 time=0.905 ms
64 bytes from 192.168.2.3: icmp_seq=4 ttl=63 time=0.815 ms
64 bytes from 192.168.2.3: icmp_seq=5 ttl=63 time=0.890 ms
64 bytes from 192.168.2.3: icmp_seq=6 ttl=63 time=0.853 ms
64 bytes from 192.168.2.3: icmp_seq=7 ttl=63 time=0.867 ms
^C
--- 192.168.2.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6026ms
rtt min/avg/max/ndev = 0.677/0.848/0.921/0.077 ms
vagrant@vm4:~$ ping google.es
ping: unknown host google.es
vagrant@vm4:~$
```

```
vagrant@r: ~$ sudo echo 1 > /proc/sys
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
vagrant@r:~$ sudo echo 1 > /proc/sys/net/ipv4/ip_forward
-bash: /proc/sys/net/ipv4/ip_forward: Permission denied
vagrant@r:~$ sudo nano /proc/sys/net/ipv4/ip_forward
vagrant@r:~$ ping google.es
PING google.es (216.58.214.163) 56(84) bytes of data:
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=1 tt
l=63 time=11.0 ms
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=2 tt
l=63 time=12.5 ms
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=3 tt
l=63 time=12.3 ms
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=4 tt
l=63 time=12.1 ms
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=5 tt
l=63 time=12.4 ms
^C
--- google.es ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/ndev = 11.859/12.262/12.512/0.236 ms
vagrant@r:~$
```

```
vagrant@vm2: ~$ sudo ifconfig eth0 down
vagrant@vm2:~$ sudo ifconfig eth1
Link encap:Ethernet  HWaddr 08:00:27:e1:dc:d4
inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe1:dc:d4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:136 errors:0 dropped:0 overruns:0 frame:0
TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:13591 (13.2 KiB)  TX bytes:8599 (8.3 KiB)

lo
Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING  MTU:65536  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

vagrant@vm2:~$ sudo route add default gw 192.168.2.1 eth1
vagrant@vm2:~$ ping google.es
ping: unknown host google.es
vagrant@vm2:~$
```

Figura 2: En esta imagen se ve como todas las máquinas intentan hacer ping a google .es pero la única que lo consigue es la máquina r (router/cortafuegos).

Para habilitar el enrutamiento usamos el comando `sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE` en la máquina router/cortafuegos. Una vez hecho eso, el resultado es el siguiente:

The image shows five terminal windows from a virtual machine environment. The top-left window (vagrant@vm3) shows the configuration of a loopback interface 'lo' and the addition of a default gateway (192.168.3.1) for the eth1 interface. It then shows successful ping tests to google.es. The top-right window (vagrant@vm1) shows the addition of a default gateway (192.168.2.1) for the eth1 interface and successful ping tests to google.es. The bottom-left window (vagrant@vm4) shows successful ping tests to google.es. The bottom-middle window (vagrant@r) shows the configuration of IP forwarding and the addition of iptables rules for NAT and masquerading, followed by successful ping tests to google.es. The bottom-right window (vagrant@vm2) shows the addition of a default gateway (192.168.2.1) for the eth1 interface and successful ping tests to google.es. All windows show the standard output of the 'ping' command, including packet statistics and round-trip times.

Figura 3: Una vez habilitado el enrutamiento, todas las máquinas tienen acceso a Internet.

- **Configurar manualmente los clientes de las redes para que se puedan conectar al servidor.**

Solo con poner la puerta de enlace y el ip forwarding estaría hecho y ya se ha hecho en el apartado anterior.

2. Servidor DHCP

Instalar un servidor DHCP en el cortafuegos. Además, se deberá modificar el fichero Vagrant, para que en lugar de establecer una IP privada, la IP se asigne mediante DHCP.

También se puede probar dejando la IP privada y comprobando el funcionamiento del servidor DHCP mediante el cliente DHCP.

El servidor DHCP deberá asignar direcciones IP a cada una de las redes internas. Además, una máquina de la segunda red tendrá que tener una dirección fija.

Tras la configuración, mostrar el estado de los prestamos realizados por el servidor DHCP.

Para instalar el servidor DHCP introducimos el siguiente comando `sudo apt-get install isc-dhcp-server`.