

Práctica 2

Jesús Rodríguez Heras
Arantzazu Otal Alberro

26 de marzo de 2019

1. Instalación de máquinas virtuales mediante Vagrant

En esta primera parte vamos a crear el entorno de trabajo, consistente en tres máquinas virtuales pertenecientes a una misma red privada. Las máquinas se tendrán que crear a partir de un mismo fichero Vagrant.

1. VM1, con IP 192.168.2.101
2. VM2, con IP 192.168.2.102
3. VM3, con IP 192.168.2.103

Las máquinas tendrán la siguiente configuración:

- nmap tiene que estar instalado en todos.
- iptables en la máquina VM1.
- ufw en la máquina VM1 (debería estar instalado por defecto).
- fwbuilder.

La instalación de los paquetes se deberá realizar mediante la provisión de Vagrant.

Para inicializar Vagrant usamos `vagrant init debian/jessie64` y luego abrimos y modificamos el archivo `Vagrantfile` de la siguiente forma:

```
1 Vagrant.configure("2") do |config|
2
3   config.vm.box = "debian/jessie64"
4   config.vm.provision :shell, path: "bootstrap.sh"
5
6   config.vm.define :vm1 do |vm1|
7     vm1.vm.box="debian/jessie64"
8     vm1.vm.hostname="VM1"
9     vm1.vm.network "private_network", ip: "192.168.2.101"
10  end
11
12  config.vm.define :vm2 do |vm2|
13    vm2.vm.box="debian/jessie64"
14    vm2.vm.hostname="VM2"
15    vm2.vm.network "private_network", ip: "192.168.2.102"
16  end
17
18  config.vm.define :vm3 do |vm3|
19    vm3.vm.box="debian/jessie64"
20    vm3.vm.hostname="VM3"
21    vm3.vm.network "private_network", ip: "192.168.2.103"
22  end
23
24 end
```

A continuación, iniciamos las tres máquinas virtuales en terminales diferentes con `vagrant up vmX` y nos conectamos a ellas mediante `vagrant ssh vmX` (siendo “X” el número de la máquina virtual comprendido entre 1 y 3).

Para instalar `nmap` en todas las máquinas usaremos el aprovisionamiento de Vagrant creando el archivo `bootstrap.sh` siguiente:

```
1 #!/usr/bin/env bash
2
3 apt-get update
4 apt-get install -y nmap
```

2. Visibilidad de las máquinas

Para los distintos ejercicios, se identifica a las máquinas como VM1, VM2 y VM3. Por comodidad, es recomendable poder usar nombres en las reglas. Para ello, se puede añadir en `/etc/hosts` una línea asociando un nombre y una IP con la siguiente sintaxis: IP NOMBRE ALIAS.

Para hacer esto, entramos en las tres máquinas virtuales y accedemos al archivo mencionado con `sudo nano /etc/hosts` y lo modificamos de la siguiente forma:

```
1 192.168.2.101 vm1
2 192.168.2.102 vm2
3 192.168.2.103 vm3
```

3. Configuraciones IPtables

3.1. Primeras pruebas

En este ejercicio se pide testear VM1 desde VM2, realizando los siguientes ejercicios:

1. Desde VM2 comprobar los puertos que VM1 tiene abiertos.

Para comprobar los puertos usamos: `nmap vm2`.

2. Prohibir el acceso por ssh.

Para ello usaremos: `sudo iptables -A INPUT -p tcp -dport 22 -j DROP`

3. Responde a las siguientes preguntas:

- **¿Qué ha pasado?**

La consola se queda bloqueada sin poder establecer conexión por ssh.

- **¿Puedo crear una nueva conexión?**

Es imposible.

- **¿La consola sigue funcionando?**

No, se queda bloqueada y no responde.

3.2. Configuración mínima

En los ejercicios siguientes, siempre debe partir de esta configuración:

- **Permitir conexiones locales.**
- **Permitir conexiones ya establecidas.**
- **Políticas por defecto de rechazar en input.**

3.3. Configurando servidor web completo

Configurar VM1 para que tenga la configuración de un servidor web, permitiendo:

- **Todos se conecten a los puertos http y https.**
- **Conexión únicamente por parte de VM2 al servidor ftp.**
- **Configurar VM1 para que sólo se pueda conectar localmente a mysql.**

3.4. Poniendo excepciones

Permitir conectar a VM1 desde VM2 y VM3 el acceso a los puertos desde 1:1000, con la excepción de que VM2 no se puede conectar por FTP.

4. UFW

Usando ufw repetir el ejercicio 3.3.