

GUÍA DE CONFIGURACIÓN DE UNA RED SIMULADA EN VIRTUALBOX

En esta guía aprenderemos a configurar 3 máquinas y un router (RouterOS) para simular una red normal y hacer diversas pruebas de seguridad sin peligro de dañar máquinas reales. Esta será la red que usaré en mi blog <https://mihackeo.blogspot.com.es/> y en todos los artículos colgados en [GitHub](#).

NOTA: Excepto la ISO de Windows, el resto de elementos y software usado es totalmente gratuito. En caso de no tener una imagen Windows, la red se puede complementar con otro Linux o probar la experiencia [Microsoft Insider](#), el cual ofrece ISOs gratuitas a usuarios que quieran probar por anticipado los S.O. para descubrir bugs. En caso de ser estudiante, la mayoría de Universidades e incluso algunos institutos ofrecen una cuenta gratuita para obtener estas imágenes.

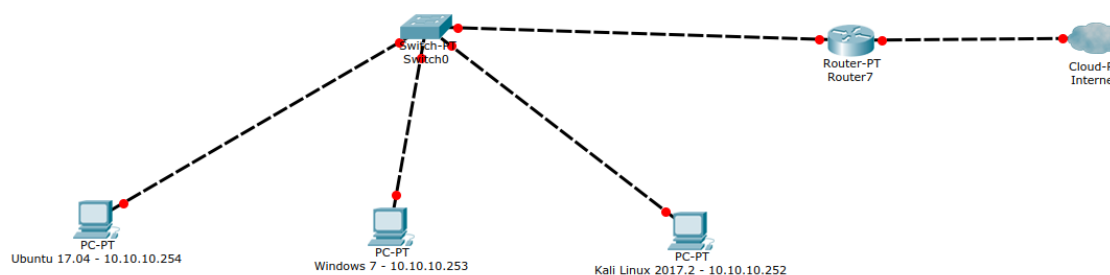
1. HACIENDO UN ESQUEMA DE RED:

En primer lugar veremos que clase de red vamos a simular. En nuestro caso, para tener una opción amplia de equipos vamos a tener:

- Un router con sistema operativo RouterOS 6.39 (Linux)
- Un equipo Kali Linux 2017.2 (Linux)
- Un equipo Windows 7
- Un equipo Ubuntu 17.04 (Linux)

Debo aclarar que para la simulación de red no es necesario el uso de un router, pero para trabajar lo más fielmente con la realidad creo que puede ser interesante tenerlo.

Por tanto la red seguiría un esquema tal y como este:



Como podeis ver, la asignación de Ips es la siguiente:

Red → 10.10.10.0/24

Router – 10.10.10.1

Ubuntu – 10.10.10.254

Windows – 10.10.10.253

Kali – 10.10.10.252

Nota: El switch modelado no es necesario en la simulación, solo es usado en el diagrama por incompatibilidades de Packet Tracer.

2. DESCARGA E INSTALACIÓN DE TODO LO NECESARIO

El siguiente paso es descargar el software de simulación y todas las imágenes ISO de los sistemas operativos. En mi caso os mostraré el software de simulación Oracle Virtualbox, ya que es gratuito y usado en muchos lugares. Existen otro tipo de software, como VMWare, pero no ofrecen los mismos servicios de forma gratuita.

VIRTUALBOX

Es un software de virtualización de máquinas, tanto en empresas como en entornos personales. Para descargarlo simplemente hay que dirigirse a <https://www.virtualbox.org/wiki/Downloads> y escoger el sistema operativo donde se desea hacer la instalación.

Para Windows y MacOS nada más pinchar se produce la descarga de un ejecutable. Para Linux abre un nuevo apartado donde pide seleccionar la distro oportuna (Ubuntu, Fedora, Debian...) y la arquitectura (32 o 64 bits). Hay que destacar que en la mayoría de los repositorios (como Debian, Ubuntu, RedHat...) aparece virtualbox, aunque personalmente aconsejo descargarlo desde la web para tenerlo actualizado.

Una vez instalado (es una instalación por defecto, de siguiente e instalar) descargaremos también el "Extension Pack" desde la misma web que el propio software:

<https://www.virtualbox.org/wiki/Downloads>

En este caso, este paquete sirve para todas las plataformas ya que se abre con Virtualbox. Una vez instalado Virtualbox, abrimos este paquete con el software e instalamos las extensiones.

Con esto tenemos Virtualbox listo para trabajar.

Podeis seguir las capturas de la web: <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2013/03/descubra-como-instalar-o-virtualbox-e-seu-pacote-de-extensoes-e-facil.html>

ROUTER EN VIRTUALBOX (ROUTEROS)

Para virtualizar un router en Virtualbox existen distintos métodos. Para mi el más sencillo y el que os presento es construir una máquina normal y convertirla en un router. Para hacer esto podríamos usar alguna distribución Linux famosa, como Debian, pero es mucho más seguro, sencillo y eficiente usar una distro Linux hecha para estos casos.

Yo he escogido RouterOS, una iniciativa de Mikrotik que se basa en tecnologías Cisco con kernel Linux y convierte un PC en un router, haciendo que la máquina únicamente cumpla la función de routear paquetes. RouterOS proporciona además firewalling, dhcp...

Para descargar su ISO solo es necesario dirigirse a <https://mikrotik.com/download/archive>, escoger la versión (en mi caso 6.39.3) y escoger el archivo .iso con la arquitectura (en mi caso 64 bits). La descarga es casi inmediata y ocupa unos 25 MB.

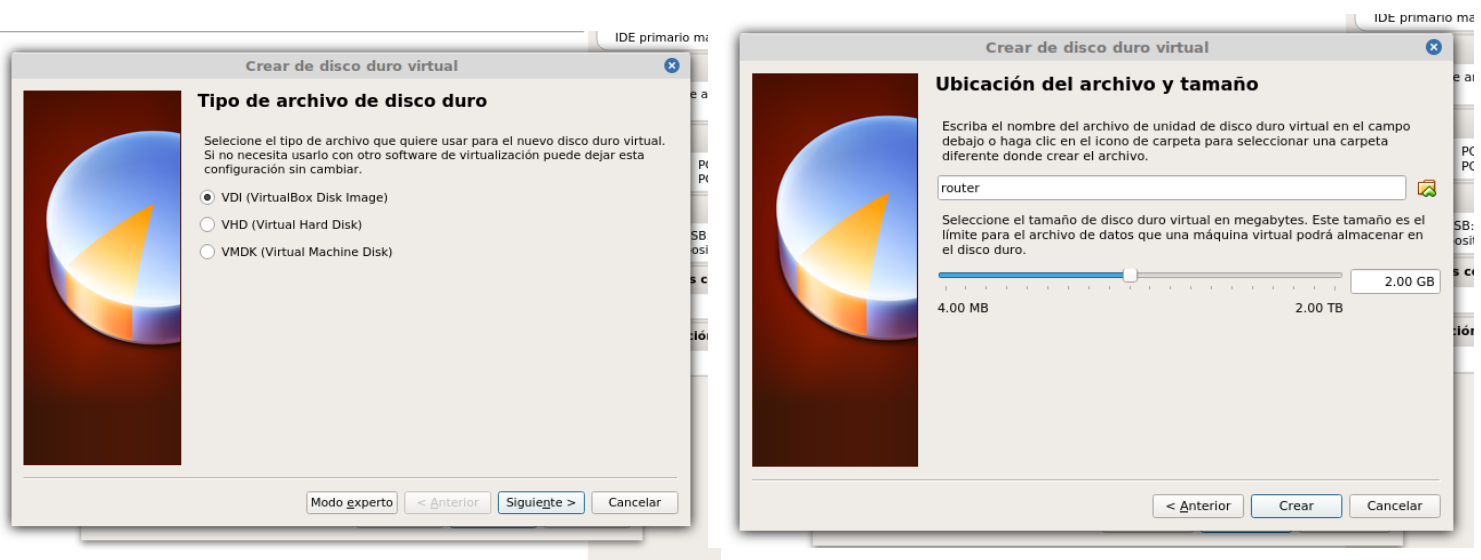
Una vez con la ISO descargada vamos a Virtualbox y pinchamos en Nueva, y nos aparecerá la siguiente pantalla:



Establecemos un nombre, tipo y versión de la misma forma que en la captura anterior. A continuación le damos un tamaño de RAM:

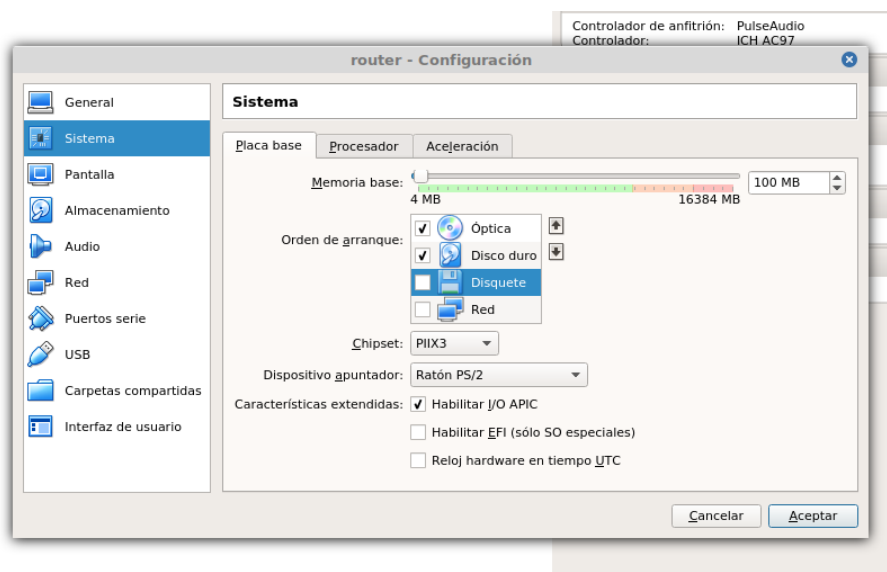


Pinchamos siguiente y crear disco. Una vez hecho esto nos pide escoger un tipo de disco de almacenamiento para Virtualbox y un tamaño:

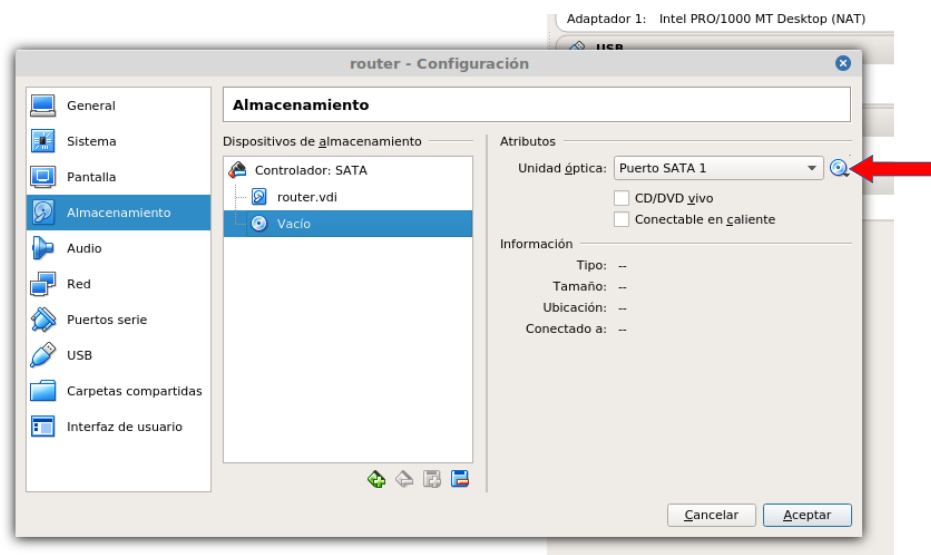


Creamos la máquina virtual y continuamos con los ajustes haciendo click en botón derecho > Configuración:

→ Establecer BIOS: Es necesario cambiar el arranque de la BIOS para instalar el disco virtual con la ISO y, posteriormente, arrancar desde el disco. Vamos a Sistema:



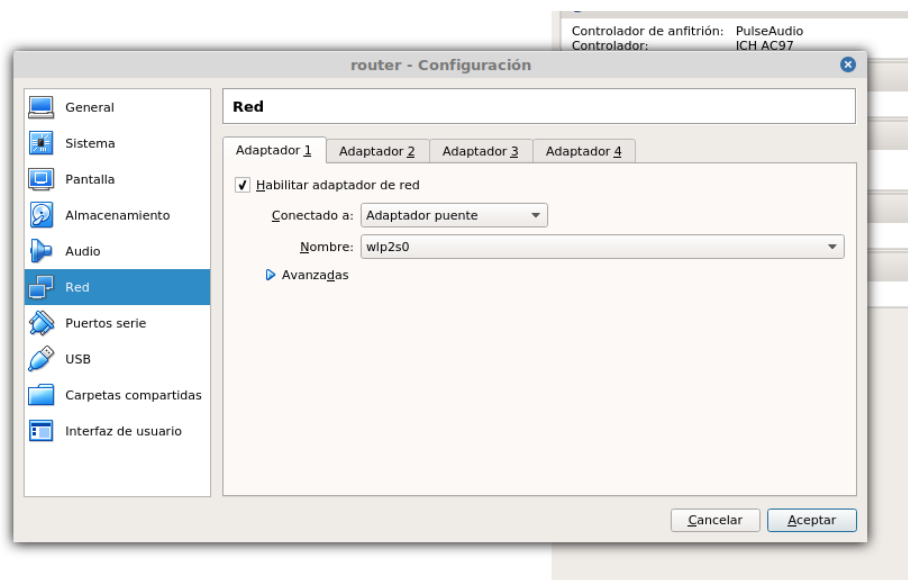
→ Introducir la ISO: Introduciremos ahora la ISO desde el apartado Almacenamiento:



Buscamos la ISO descargada y la introducimos.

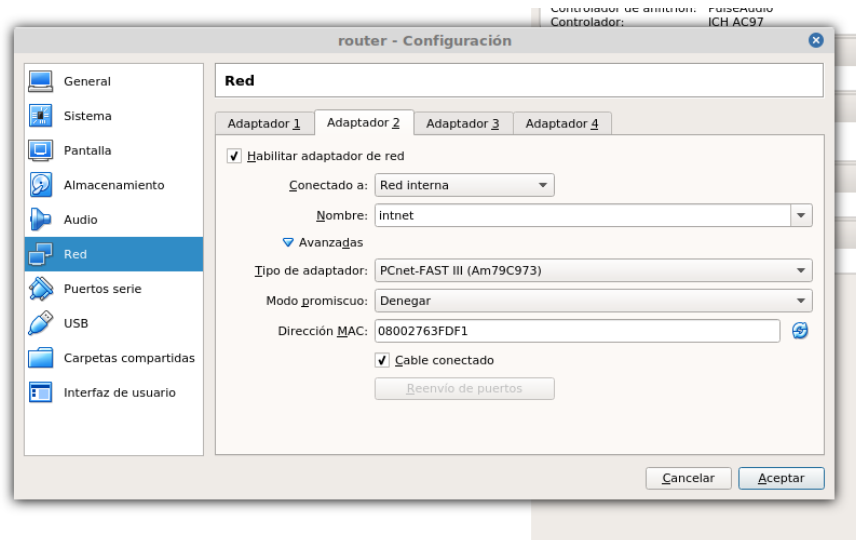
→ Establecemos las interfaces de red: Para todos los equipos necesitaremos 2 interfaces de red, una que salga directa a Internet y otra a la red que vamos a montar. Debemos recalcar que los demás equipos bastaría que tuvieran una sola interfaz ya que el router haría de puente, pero lo haremos así por si en algún momento queremos tener conexión sin el router. Para ello estableceremos dos interfaces:

- Adaptador 1: salida a Internet. En Virtualbox se modela como un Adaptador-Puente, es decir, ocupa una IP del rango en la misma red de la máquina donde corre la máquina virtualizada.



Hay que cercionarse de que el adaptador de red esté habilitado y la interfaz de red sea la correcta.

- Adaptador 2: red que queremos montar. Se modela en Virtualbox como Red Interna.



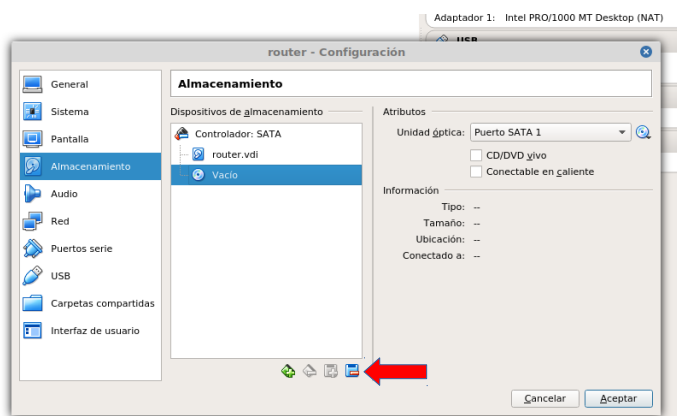
Muy importante: asegurarse que la red se llame igual en todos los dispositivos (por defecto intnet) y muy recomendable usar 2 MAC distintas en ambos adaptadores (simula tener dos tarjetas de red).

Una vez configurado todo, el siguiente paso es ejecutar la máquina. La seleccionamos y damos iniciar.

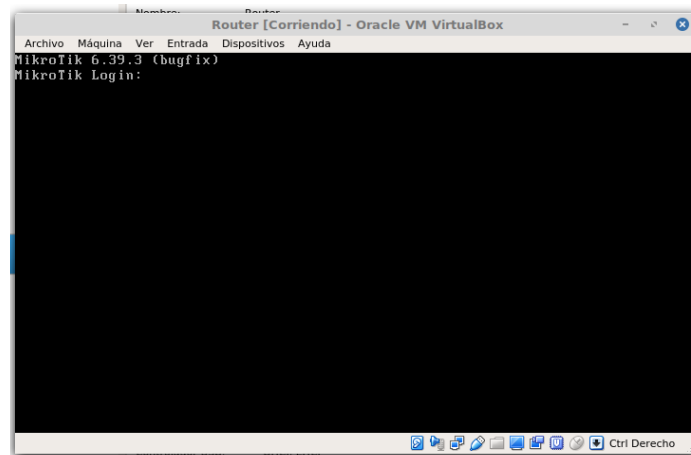
Para la instalación del sistema operativo, en primer lugar arrancará la ISO que procederá a la instalación. Una vez cargados los drivers y los distintos elementos, RouterOS te dará la opción de introducir que paquetes quieres que se instalen en la máquina. Por defecto pulsaremos la tecla A (de all) que añadirá todos los paquetes. Posteriormente pulsaremos I (de install) para instalar. Nos preguntará si deseamos continuar y pulsaremos Y (de yes). Se producirá por tanto la instalación y cuando termine (esto dura unos pocos segundos) pedirá pulsar Enter para rebootear. Nosotros NO lo haremos, sino que cerraremos la virtualización de la máquina (cerrar el programa donde está ejecutandose la máquina). Aquí Virtualbox nos ofrece 3 alternativas:

- Guardar el estado: dejar la máquina en el estado en el que se encuentra pero cerrando la virtualización. Cuando iniciemos la máquina de nuevo, la máquina se iniciará en ese estado.
- Enviar señal de apagado: se corresponde a pulsar el botón de encendido/apagado del PC.
- Apagar la máquina: equivale a quitarle la corriente y apagarla.

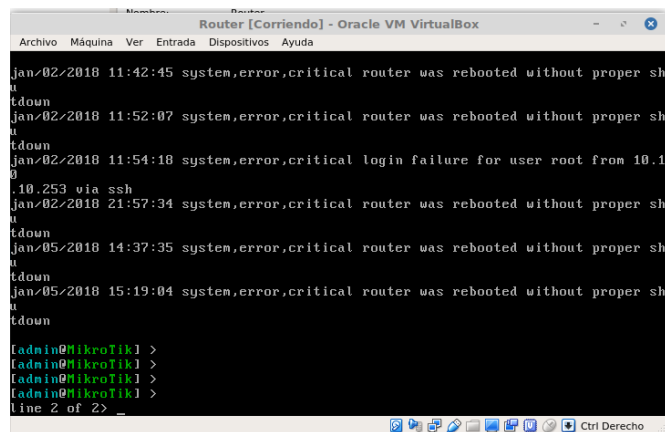
Optaremos por esta última opción. Una vez echo, la virtualización se parará y podremos quitar la ISO. Importante tener seleccionado el disco.



Ahora iniciaremos la máquina. En primer lugar nos pedirá usuario y contraseña, que por defecto es “admin” (sin comillas, claro) y sin contraseña.



Una vez logueado nos sale una terminal.



Nota: Virtualbox captura el ratón en la máquina y hay veces que este no funciona en nuestro escritorio. Para solucionarlo pulsamos CTRL de la derecha y listo.

Una vez ya tenemos el router con su S.O. configuraremos las partes básicas: las IPs del router y la contraseña para loguearnos (es de locos no cambiar passwords por defecto).

- Password: RouterOS lo caracterizaría por la sencillez de comandos. Si pulsamos Tabulador en la shell podemos ver todos los comandos que se pueden usar en esa opción. Si lo pulsamos en la opción raíz vemos que hay una llamada password. Escribimos pues password en la shell y damos Enter: nos pregunta en primer lugar la contraseña actual (recordamos que no tiene contraseña) y luego nos pregunta la nueva y su confirmación. Una vez introducidas ya tenemos la nueva contraseña para loguearse.
- IPs: Tendremos que entrar en la opción ip, y una vez allí la opción address. Para establecer la ip debemos recordar que la interfaz 1 (para RouterOS ether1) es la salida a nuestra red interna “real” y la interfaz 2 (ether2) es la red virtualizada. Entonces para añadir las IPs escribimos los comandos:

```
add address=192.168.1.254 netmask=255.255.255.0 network=192.168.1.0 broadcast=192.168.1.255 interface=ether1
add address=10.10.10.1 netmask=255.255.255.0 network=10.10.10.0 broadcast=10.10.10.255 interface=ether2
```

NOTA → Mi red “real” es 192.168.1.0/24. Podeis comprobarlo vosotros mediante ipconfig (Windows) o ifconfig (Unix).

Ya tenemos nuestro router puesto en marcha. Podemos comprobar sus direcciones con print desde ip/address.

LINUX EN VIRTUALBOX

Obtener ISO de Ubuntu: <https://www.ubuntu.com/download/desktop>

Obtener ISO de Kali: <https://www.kali.org/downloads>

El mecanismo de creación de la máquina es el mismo que en el router, cambiando la RAM y el disco:

RAM = entre 2 y 4 GB

Disco = 25 GB

Las interfaces de red, el montaje de la ISO y la BIOS se hace exactamente igual que en el Router y la instalación del S.O. es muy sencilla:

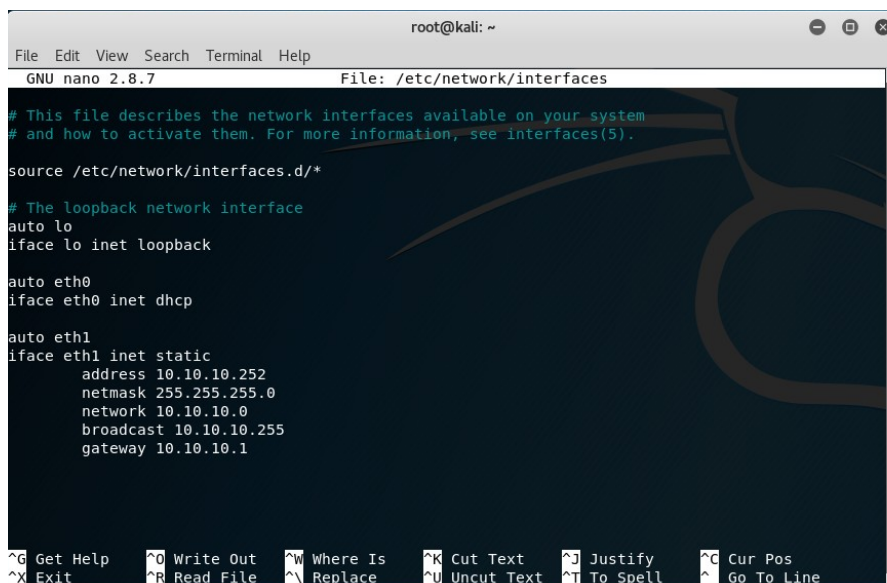
Ayuda: [Ubuntu](#) | [Kali](#)

Para que la máquina se vea correctamente en Virtualbox instalaremos los Guest Additions:

<https://forums.virtualbox.org/viewtopic.php?t=15679>

Ahora simplemente configuramos las IPs y las interfaces. Para ello abrimos un terminal y escribimos como root:

nano /etc/network/interfaces

A screenshot of a terminal window titled 'root@kali: ~'. The terminal shows the nano text editor editing the file '/etc/network/interfaces'. The content of the file is as follows:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

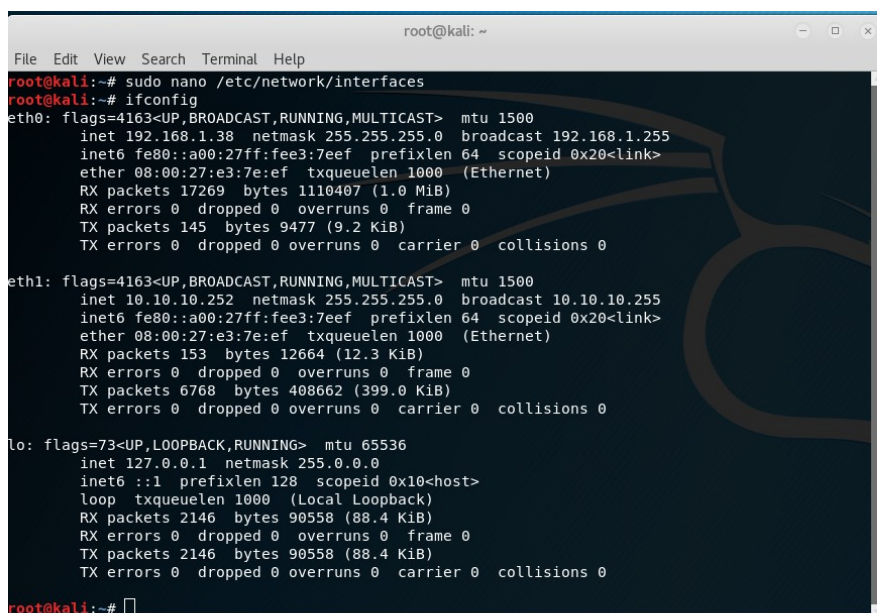
auto eth1
iface eth1 inet static
    address 10.10.10.252
    netmask 255.255.255.0
    network 10.10.10.0
    broadcast 10.10.10.255
    gateway 10.10.10.1
```

The bottom of the terminal shows the nano editor's command palette with various shortcuts like ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^N Replace, ^U Uncut Text, ^T To Spell, and ^_ Go To Line.

Usamos CTRL+O para guardar y CTRL+X para salir.

Para Ubuntu es exactamente igual cambiando **address 10.10.10.254**.

Finalizamos haciendo un reboot y, al reiniciarse, comprobamos las interfaces con ifconfig:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# sudo nano /etc/network/interfaces  
root@kali:~# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.38 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::a00:27ff:fee3:7eef prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:e3:7e:ef txqueuelen 1000 (Ethernet)  
    RX packets 17269 bytes 1110407 (1.0 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 145 bytes 9477 (9.2 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.10.10.252 netmask 255.255.255.0 broadcast 10.10.10.255  
    inet6 fe80::a00:27ff:fee3:7eef prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:e3:7e:ef txqueuelen 1000 (Ethernet)  
    RX packets 153 bytes 12664 (12.3 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6768 bytes 408662 (399.0 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 2146 bytes 90558 (88.4 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 2146 bytes 90558 (88.4 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
root@kali:~#
```

WINDOWS EN VIRTUALBOX

Esta parte la reduzco ya que la mayoría de personas no tienen la ISO ni usan Windows en hacking.

La creación y configuración de la máquina es exactamente igual a las anteriores, aunque yo personalmente le daría 10 GB más de disco por si las moscas.

Instalación Windows 7: <https://www.softzone.es/manuales-software-2/instalar-windows-7-manual-de-instalacion-de-windows-7/>

Necesitamos configurar la IP de la red interna virtualizada. Para ello podremos seguir este manual:

http://www.dlink.com/es/es/support/faq/adapters/fr_comment-configurer-une-adresse-ip-fixe-sur-mon-ordinateur

En nuestro caso SÓLO configuramos la interfaz 2 de red con los datos de la red 10.10.10.0.

3. VERIFICANDO LA RED

En este paso lo único que haremos en cada máquina es comprobar:

- Salida a Internet
- Salida a red
- Acceso a red virtual
- Acceso entre equipos
- Verificar equipos en la red.

Para verificar los 4 primeros he hecho 3 pings a diferentes IPs

8.8.8.8 → DNS de Google para comprobar salida a Internet

10.10.10.1 → Router para comprobar acceso entre equipos y acceso a red virtual

192.168.1.36 → Equipo en red para comprobar el acceso a esta

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# 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=54 time=44.5 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=53.9 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=54 time=46.0 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=54 time=149 ms  
^C  
--- 8.8.8.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 44.592/73.559/149.635/44.066 ms  
root@kali:~# ping 10.10.10.1  
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.  
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.310 ms  
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.696 ms  
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.206 ms  
^C  
--- 10.10.10.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2028ms  
rtt min/avg/max/mdev = 0.206/0.404/0.696/0.210 ms  
root@kali:~# ping 192.168.1.36  
PING 192.168.1.36 (192.168.1.36) 56(84) bytes of data.  
64 bytes from 192.168.1.36: icmp_seq=1 ttl=64 time=9.49 ms  
64 bytes from 192.168.1.36: icmp_seq=2 ttl=64 time=2.00 ms  
64 bytes from 192.168.1.36: icmp_seq=3 ttl=64 time=1.47 ms  
64 bytes from 192.168.1.36: icmp_seq=4 ttl=64 time=3.48 ms  
64 bytes from 192.168.1.36: icmp_seq=5 ttl=64 time=4.03 ms  
^C  
--- 192.168.1.36 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4005ms  
rtt min/avg/max/mdev = 1.471/4.098/9.494/2.856 ms  
root@kali:~#
```

Posteriormente un nmap a la red virtualizada para comprobar equipos en ella:

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nmap 10.10.10.0/24  
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-02 14:00 CET  
Nmap scan report for 10.10.10.1  
Host is up (0.00021s latency).  
Not shown: 994 closed ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
80/tcp    open  http  
2000/tcp  open  cisco-scp  
8291/tcp  open  unknown  
MAC Address: 08:00:27:35:03:96 (Oracle VirtualBox virtual NIC)  
  
Nmap scan report for 10.10.10.252  
Host is up (0.0000050s latency).  
All 1000 scanned ports on 10.10.10.252 are closed  
  
Nmap done: 256 IP addresses (2 hosts up) scanned in 8.06 seconds  
root@kali:~#
```

¡Y la red ya está lista!