CONFIGURACIÓN DE LABORATORIO VIRTUAL DE SEGURIDAD

Práctica 1

Informe de prácticas

1. **Indica las características de cada una de las máquinas virtuales utilizadas para montar el entorno de trabajo virtual: sistema operativo y arquitectura, número de procesadores, cantidad de memoria RAM asignada, velocidad de transmisión del adaptador de red.**

   Si nos fijamos en la información general de cada una de las máquinas virtuales podemos obtener toda la información que se requiere:

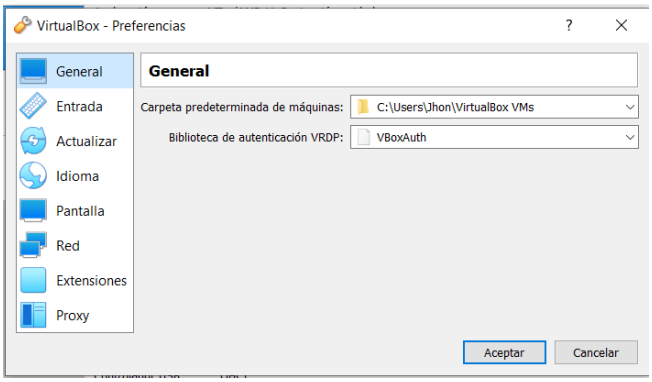
   - ***alice***: Esta máquina utiliza la distribución Linux: **Ubuntu** (32-bit). En cuanto al número de procesadores, tiene 1 **procesador,** 512 MB de memoria RAM asignada y una velocidad de 1000 Mbit/s de transmisión.

   - ***bob***: Por otra parte, esta máquina utiliza el sistema operativo **Debian** (32-bit). En cuanto al número de procesadores, tiene 1 **procesador,** 384 MB de memoria RAM asignada y una velocidad de 1000 Mbit/s de transmisión.

   - ***mallet***: Finalmente, esta máquina utiliza nuevamente la distribución Linux: **Ubuntu** (32-bit). En cuanto al número de procesadores, tiene 1 **procesador,** 512 MB de memoria RAM asignada y una velocidad de 1000 Mbit/s de transmisión.

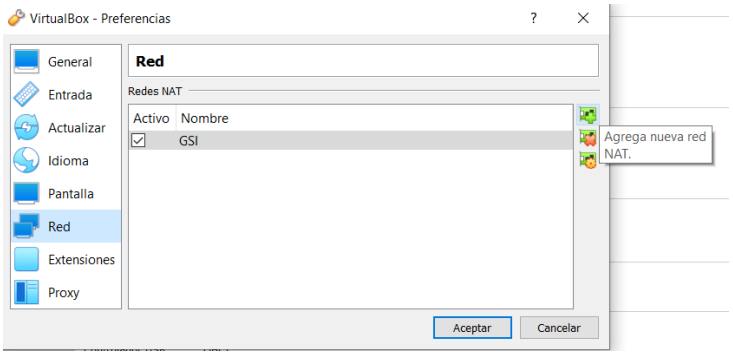
2. **Configura el entorno de virtualización y las máquinas virtuales para que las tres máquinas se encuentren dentro de la Red NAT 10.0.2.0/24 de nombre GSI". Documenta todos los pasos realizados.**

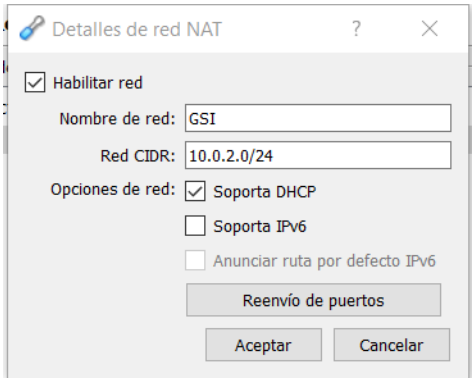
   Lo primero que debemos hacer es ir a *VirtualBox – Preferencias* y una vez ahí, nos dirigimos al apartado de *Red*.



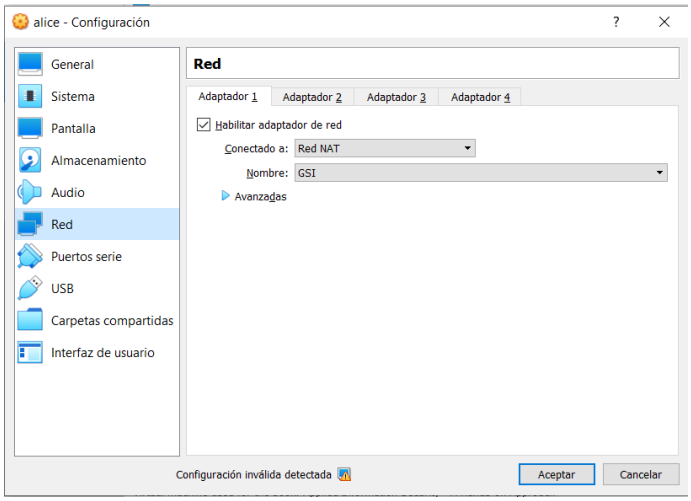
Procedemos a *Agregar nueva red NAT*, hacemos doble clic sobre la red creada "*NatNetwork*" y nos aparecen los detalles de la red NAT. Editamos el nombre de la red, en este caso el nombre será "GSI" y como dirección configuramos la que nos dan: 10.0.2.0/24.

En cuanto a las opciones de red, dejamos marcado *Soporta DHCP* para que proporcione direccionamiento IPv4 a todas las máquinas.





Una vez creado la red NAT con el direccionamiento adecuado, tenemos que configurar cada una de las 3 máquinas. Para ello, vamos a *Configuración* y en el apartado *Red* en la opción de *Conectado a* seleccionamos Red NAT y en *Nombre* nos aparecerá directamente "GSI". Finalmente aceptamos y ya tendremos las 3 máquinas dentro de la Red creada.



3. **Dibuja un diagrama de red lo más detallado posible de la red que forman las tres máquinas virtuales: *alice*, *bob*, *mallet*. Para cada máquina proporciona su dirección MAC, dirección IPv4 y máscara de red. Indica la dirección de red en formato IPv4 de la red en la que se encuentran las máquinas y la puerta de enlace (*gateway*) de cada una de ellas. ¿Es la misma? Justifica tu respuesta.**

   Para encontrar las diferentes direcciones y la máscara de red utilizamos el comando *ifconfig* el cual muestra la información de configuración de todas las interfaces de red y la dirección IP asociada.

DNS

Dirección MAC: **52:54:00:12:35:00**

Dirección IPv4: **10.0.2.2**

Máscara de red: **255.255.255.0**

mallet

Dirección MAC: **08:00:27:62:45:e6**

Dirección IPv4: **10.0.2.5**

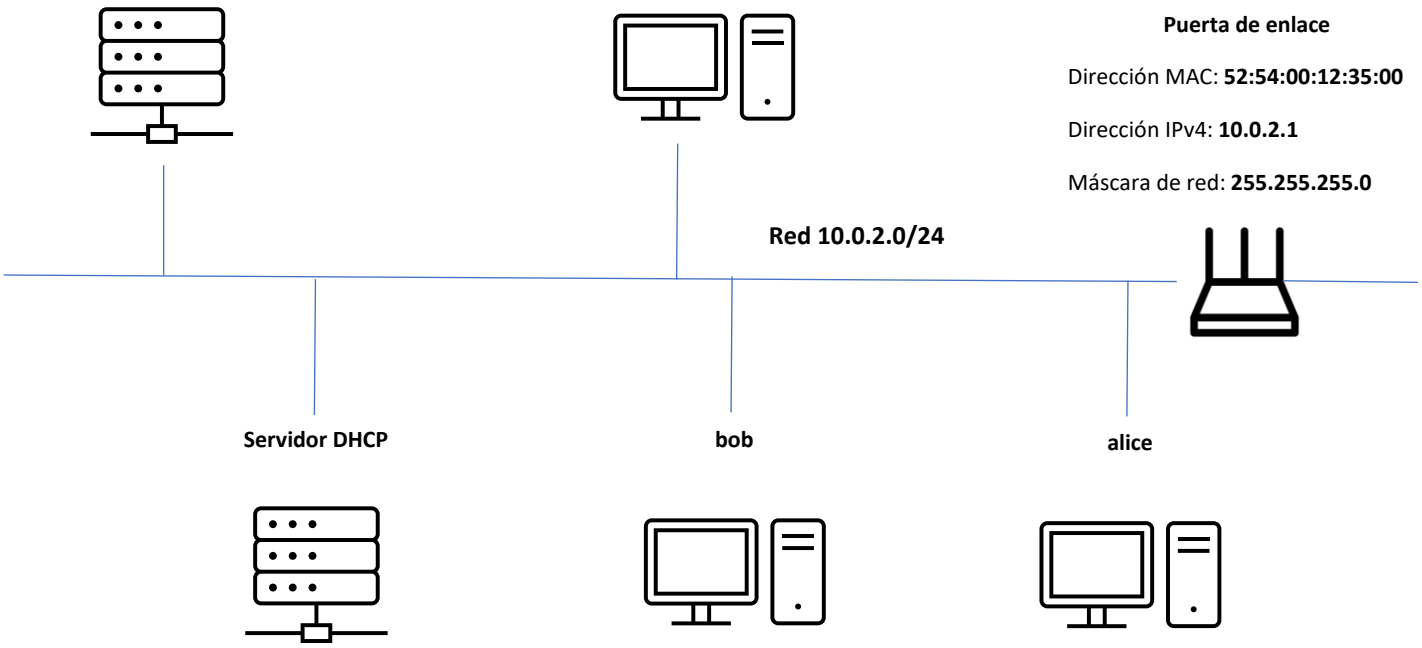
Máscara de red: **255.255.255.0**

Puerta de enlace

Dirección MAC: **52:54:00:12:35:00**

Dirección IPv4: **10.0.2.1**

Máscara de red: **255.255.255.0**



Dirección MAC: **08:00:27:f1:b3:7b**

Dirección IPv4: **10.0.2.3**

Máscara de red: **255.255.255.0**

Dirección MAC: **08:00:27:63:40:98**

Dirección IPv4: **10.0.2.15**

Máscara de red: **255.255.255.0**

Dirección MAC: **08:00:27:48:f3:46**

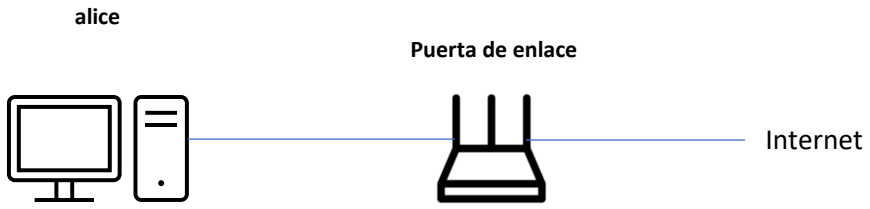
Dirección IPv4: **10.0.2.4**

Máscara de red: **255.255.255.0**

Para encontrar la dirección de la puerta de enlace utilizamos el comando ***route***. Observamos que las 3 máquinas tienen la misma puerta de enlace: 10.0.2.1 debido a que se encuentran en la misma Red NAT (GSI).

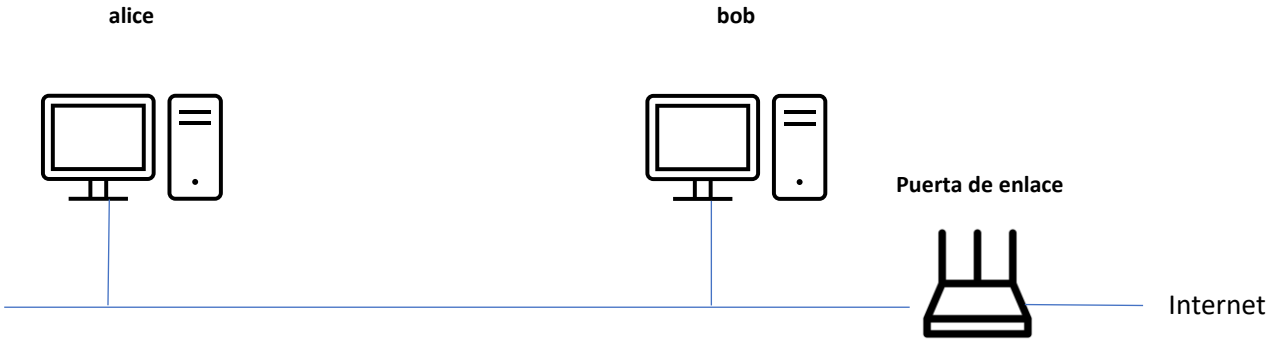
4. **Indica con tus palabras cuál es la diferencia, en el sistema de virtualización utilizado, entre el modo NAT, red NAT y red interna; para entenderlo mejor, proporciona un ejemplo gráfico de cada uno de los modos de funcionamiento.**

El modo **NAT** permite establecer conexiones desde las máquinas virtuales a Internet, pero no permite la comunicación entre distintas máquinas virtuales.



El modo **red interna** en cambio, permite la conexión entre las máquinas virtuales, pero no permite la salida a Internet desde las máquinas virtuales.

El modo **Red NAT** permite que las máquinas virtuales se comuniquen entre sí lo que permite crear entornos complejos de comunicación y al mismo tiempo permite asignar direcciones IP de manera que pueda tener conexión a Internet.



5. **Realiza pruebas para comprobar que las máquinas virtuales se comunican entre sí a nivel de red (capa 3 del modelo de referencia OSI). Para ello, puedes utilizar el comando *ping.* Documenta la información necesaria que justifique que las máquinas tienen comunicación a nivel de red.**

   Comenzamos modificando el fichero de configuración para cada una de las 3 máquinas. Para ello, deberemos tener privilegios administrativos y seguido, utilizar el comando **nano /etc/network/interfaces**.

   Añadimos la definición para cada una de las interfaces de red (eth0, eth3 y eth4). Comenzamos con una línea que incluya la palabra *auto*, seguida por el nombre de las interfaces de red que vayamos a definir. A continuación, incluimos una segunda línea donde se especifica que la configuración se realizará por medio de DHCP, es decir, que aceptamos direcciones IP propuestas por un servidor DHCP.

   Guardamos los cambios y procedemos a reiniciar los servicios de red para que las modificaciones en la configuración de las interfaces de red sean aplicadas. Para ello podemos utilizar el comando **/etc/init.d/networking restart**. Otra manera de hacerlo consiste en habilitar y deshabilitar las tarjetas de red mediante los comandos **ifdown** e **ifup**.

   Finalmente, utilizamos el comando **ping** para comprobar la comunicación entre las máquinas.

   1. Realizamos un *ping* de alice hacia bob y hacia mallet determinando un número específico de peticiones:

ping 10.0.2.15 -c 5

```
root@alice:/home/alice# ping 10.0.2.15 -c 5
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.548 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.811 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=1.21 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.444 ms

--- 10.0.2.15 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.444/0.848/1.228/0.326 ms
```

ping 10.0.2.5 -c 5

```
root@alice:/home/alice# ping 10.0.2.5 -c 5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=0.665 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=1.38 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=1.44 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=1.29 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=1.44 ms

--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.665/1.246/1.448/0.299 ms
```

2. Realizamos un *ping* de bob hacia alice y hacia mallet determinando un número específico de peticiones:

ping 10.0.2.4 -c 5

```
bob:/home/bob# ping 10.0.2.4 -c 5
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.967 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=2.39 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=3.09 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=1.17 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=1.62 ms

--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 0.967/1.849/3.090/0.790 ms
```

ping 10.0.2.5 -c 5

```
bob:/home/bob# ping 10.0.2.5 -c 5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=2.76 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=1.13 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=1.15 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=1.69 ms

--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 1.092/1.567/2.760/0.636 ms
```

3. Realizamos un *ping* de mallet hacia alice y hacia bob determinando un número específico de peticiones:

ping 10.0.2.4 -c 5

```
root@mallet:/home/mallet# ping 10.0.2.4 -c 5
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.437 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=1.64 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.532 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=1.22 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=1.61 ms

--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.437/1.090/1.649/0.519 ms
```

ping 10.0.2.15 -c 5

```
root@mallet:/home/mallet# ping 10.0.2.15 -c 5
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.667 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.503 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.38 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.858 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=1.11 ms

--- 10.0.2.15 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.503/0.905/1.386/0.317 ms
```

6. **¿Es posible inferir el sistema operativo de cada una de las máquinas a través del valor del TTL (*Time To Live*) del paquete que devuelven las máquinas después de recibir una petición de tipo ICMP(8)? Justifica tu respuesta. El siguiente enlace puede resultarte de utilidad: https://subinsb.com/default-device-ttl-values/**

   El tiempo de vida (TTL) hace referencia a la cantidad de tiempo o "saltos" que se ha establecido que un paquete debe existir dentro de una red antes de ser descartado por un enrutador, el cual enviará un mensaje ICMP al host origen.

   Los valores TTL son diferentes para los Sistemas Operativos, por lo que podemos determinar el Sistema Operativo basándonos en su valor específico. Estos son algunos valores por defecto de TTL de diferentes Sistemas Operativos:

   - Linux/Unix: 64
   - Windows: 128
   - MacOS: 64
   - Solaris/AIX: 254

   Si nos fijamos en las imágenes anteriores correspondientes con la realización de distintos *ping*, el valor devuelto para el TTL es siempre 64 el cual se corresponde con Linux/Unix y si recordamos, las 3 máquinas tenían una distribución de Linux, por lo que efectivamente, sí que es posible inferir el S.O de una máquina a través del TTL.

7. **Desde la máquina "*mallet*", utiliza la herramienta "*nmap*" para realizar un descubrimiento de los host que se encuentren en su mismo segmento de red pero sin escanear ningún servicio TCP/UDP. ¿Qué protocolo te parece más adecuado para ello, ARP o ICMP? Justifica tu respuesta.**

   Utilizamos el comando **sudo nmap 10.0.2.0/24 -n -sP** , donde especificamos la red que queremos escanear y añadimos los siguientes flags: *-n* para que cada vez que nmap detecte una máquina no trate de averiguar el nombre de esa máquina (No realizar resolución de nombres) y *-sP* para realizar un descubrimiento de sistemas mediante un sondeo ping, y luego emitir un listado de los equipos que respondieron al mismo. Una vez realizado el descubrimiento obtenemos los siguientes host:

   - **10.0.2.1/24** → Puerta de enlace
   - **10.0.2.2/24** → DNS de VirtualBox
   - **10.0.2.3/24** → Servidor DHCP
   - **10.0.2.4/24** → alice
   - **10.0.2.5/24** → mallet
   - **10.0.2.15/24** → bob

En cuanto al protocolo más adecuado, si realizamos un análisis de tráfico de red mediante ***tcpdump***, el protocolo que aparece en la mayoría es el protocolo **ARP** el cual se encarga de vincular una dirección MAC o dirección física, con una dirección IP o dirección lógica. Por su parte, el protocolo ICMP es un protocolo en la capa de red que utilizan los dispositivos de red para diagnosticar problemas de comunicación en la red.

Por lo tanto, considero más adecuado utilizar el protocolo ARP, ya que al estar analizando una única red este protocolo resulta más rápido y fiable.

8. Indica los problemas que te has encontrado y cómo los has resuelto.

- Cuál era el fichero que había que modificar para configurar las interfaces de red. Resuelto en la clase de laboratorio.
- Recordar el funcionamiento de diferentes protocolos tales como ARP o ICMP. Repaso de protocolos de redes.
- Uso correcto del comando nmap. En la clase de laboratorio se explicó y recordó su uso y funcionamiento. Además de buscar por Internet para qué servían ciertos flags.