UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Diseño de un entorno de trabajo seguro empleando varias máquinas virtuales VMWare simultáneamente.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Informática

Autora: Mª de la Almudena Igualá Villarroya

Tutores: Ismael Ripoll Ripoll, Hector Marco Gisbert

2016/2017

Resumen

En este proyecto se afronta la problemática de la seguridad de los sistemas en la época actual, a la vez que se aporta una posible solución mediante aislamiento de procesos. Con el fin de conseguir un sistema seguro, se plantea una solución básica inspirada en el sistema Qubes OS utilizando VirtualBox como sistema de virtualización y Netfilter como firewall, aplicándose este desde la máquina anfitriona con sistema Ubuntu.

Palabras clave: VirtualBox, Netfilter, Qubes OS, firewall, Linux, ciberseguridad.

Abstract

This project tackles the problems of system security in the current era, while providing a possible solution through process isolation. In order to get a secure system, a basic solution inspired by the system Qubes OS is propounded, using VirtualBox as a virtualization system and Netfilter as a firewall, being applied this from the host machine with Ubuntu system.

Keywords : VirtualBox, Netfilter, Qubes OS, firewall, Linux, cybersecurity.

Tabla de contenidos

1. Introducción

Actualmente el ordenador es utilizado por usuarios de todo tipo y su uso es de lo más variado, por ejemplo, se puede utilizar como herramienta de trabajo, como instrumento de ocio, para realizar compras, etc. Esto implica una serie de riesgos para la seguridad, pues un ataque de tipo ransomware, por ejemplo, en el equipo de trabajo podría significar la pérdida de horas y horas de trabajo. Otro ataque bastante común sería el robo de tarjetas de crédito y de toda la información que haya al alcance. ¿Cómo se podría evitar esto? Hay diversas soluciones, además de ser extremadamente cuidadoso de qué páginas se visitan, en qué lugares se compra, qué programas, y de dónde, se descargan etc., también están los firewalls, los antivirus, los anti-malware, etc. Igualmente, estos métodos no son infalibles, pues cada día aparecen más virus y malware y cada vez son más sofisticados y difíciles de neutralizar.

Por este motivo, hoy en día las aproximaciones convencionales de seguridad como son los antivirus y los firewall no son una barrera suficiente potente como para neutralizar a los atacantes más sofisticados. Actualmente es común que los atacantes comprueben si sus malware son reconocibles por antivirus y anti-malware, y de ser así, editan y mejoran el código hasta que no sea reconocible. Normalmente los antivirus son actualizados cuando sus programadores descubren una nueva amenaza y la forma de neutralizarla, cosa que no suele ocurrir hasta unos días después de que el ataque se esté realizando a nivel masivo, y la actualización llega tarde para los que ya hayan visto su información comprometida. Es por estos motivos que se podría decir que el firewall y el antivirus no son una barrera lo suficientemente eficaz.

El objetivo de este proyecto es explicar e implementar otro tipo de solución, una solución basada en el aislamiento de procesos, de forma que se puedan paliar los daños producidos por un hipotético ataque. La solución planteada se basaría en un sistema MSL, siglas de *Multiple Single-Level*, en este sistema, cada proceso es colocado en un dominio separado no confiable, de forma que no haya comunicación entre los distintos procesos. Este tipo de separación implica separación física habitualmente, es decir, una máquina física para cada proceso. En este caso, la separación se realizará mediante máquinas virtuales, colocadas cada una en una red virtual distinta. De esta forma, cada una de las máquinas se dedicará únicamente a una o un pequeño grupo de tareas similares, ya que eso facilita la tarea de securizar el entorno. Por ejemplo, una máquina dedicada únicamente a la lectura de correo, pues es uno de los principales vectores de ataque, otra máquina dedicada al trabajo, otra para ocio y tiempo libre (videojuegos, navegar por internet, descarga y/o visionado de películas y series), etc.

El motivo por el que me decidí por hacer este proyecto en lugar de cualquiera de los otros propuestos es porque la ciberseguridad, en mi opinión, es un tema de

actualidad, y que es un entorno cambiante y en constante evolución, además de que es importante aprender cómo securizar nuestros equipos, pues hoy en día, toda nuestra vida está en un ordenador, la declaración de la renta, nuestros datos médicos, la información bancaria, etc. Es por este motivo que considero muy importante aprender a protegerse frente a posibles ataques, de forma que los riesgos sean mínimos. Otro motivo que me ha llevado a interesarme en este proyecto es el uso del firewall, ya que es una herramienta muy útil pero que no siempre se sabe utilizar y configurar, y, de esta forma podría ampliar mi conocimiento sobre firewalls, puramente teórico, y llegar a realizar una implementación de un firewall más completo y funcional.

Finalmente, pese a que en el título se especifica el uso de VMware, se decidió junto con los tutores utilizar VirtualBox, pues es el entorno que se ha venido utilizando a lo largo de toda la carrera y, por lo tanto, era del que más conocimientos poseía.

2. Estado del arte

En la línea de estudio de este proyecto podemos encontrar Qubes OS, un sistema operativo orientado a la seguridad. Qubes OS es gratuito y de código abierto, lo que significa que cualquiera es libre de utilizar, copiar y/o cambiarlo como desee, además cualquiera puede contribuir a su mejora ya sea añadiendo mejoras de código, o reportando errores.

Con Qubes OS se implementa seguridad por compartimentación, lo que permite al usuario separar las distintas actividades que se realizan comúnmente en un ordenador en varias máquinas virtuales aisladas. Esta aproximación permite al usuario mantener las actividades separadas en distintas máquinas, de esta forma si la seguridad de una de ellas se viera comprometida, no afectaría a las otras. Un ejemplo sencillo sería: si entrase algún tipo de malware a través del correo, y se usa la misma máquina para consultar el correo y para realizar compras por internet, la seguridad de los datos bancarios podría verse comprometida. Con el sistema de Qubes se podrían separar estas dos actividades en máquinas distintas, de forma que por muy atacado que se viera nuestro correo, no afectaría a nuestros datos bancarios, al no encontrarse en esa máquina. De esta forma con Qubes el usuario puede hacerlo todo en la misma máquina física sin tener que preocuparse tanto por posibles ciberataques.

La diferencia entre el uso de máquinas virtuales comunes y Qubes OS es que los programas como VirtualBox o Vmware, conocidos como hipervisores tipo 2, se ejecutan sobre los sistemas habituales como son Windows o iOS y estas máquinas virtuales son tan seguras como lo sea el sistema anfitrión, es decir, si el sistema se viera comprometido, las máquinas virtuales alojadas en él también se verían comprometidas.

Sin embargo, Qubes OS utiliza un hipervisor tipo 1 llamado Xen. En lugar de ejecutarse sobre un sistema operativo, los hipervisores tipo 1 se ejecutan directamente sobre le hardware, lo que significa que un atacante tendría que infectar el hipervisor en sí para poder comprometer el sistema completo.

Qubes hace esto de forma que varias máquinas virtuales ejecutándose sobre un hipervisor tipo 1 puedan utilizarse de forma segura como un sistema operativo totalmente integrado. Por ejemplo, se pueden poner todas las aplicaciones en el mismo escritorio con un borde de colores indicando la fiabilidad de cada máquina virtual. Además, también permite realizar operaciones de copiado y pegado entre máquinas de forma segura, además de garantizar una conexión segura entre las máquinas e internet.

Utilizar un ordenador distinto para algunas tareas puede ser también una opción mucho más segura a la hora de tratar con datos sensibles, sin embargo, siguen habiendo algunos riesgos que se deberían tener en cuenta. A continuación se lista

una serie de pros y contras de utilizar una separación física respecto a un sistema basado en Qubes:

- Pros
    - La separación física no depende de un hipervisor. Aunque sea complicado acceder al hipervisor de Qubes, si alguien consiguiera hacerlo, podría adquirir control sobre todo el sistema.
    - La separación física como medida de seguridad en sí. Por ejemplo, se podría considerar seguro guardar el "ordenador seguro" en una caja fuerte cuando se lleva el "ordenador inseguro".
- Contras
    - La separación física puede resultar complicada y cara, pues se debería obtener una máquina para cada nivel de seguridad de deseáramos tener.
    - Generalmente, no hay una forma segura de transferir datos entre distintas máquinas que utilicen sistemas operativos convencionales.
    - Los ordenadores separados físicamente con sistemas operativos convencionales siguen siendo vulnerables a los ataques más habituales debido a su naturaleza monolítica.

3. Análisis del entorno

Disponemos de una máquina host con Ubuntu 16.04.1 LTS, 8 GB de RAM y 930 GB de disco duro. Debido a las necesidades de las máquinas que se describirán a continuación, no se podrán conectar todas las máquinas de forma simultánea por problemas de memoria RAM, se contempla la posibilidad de hacer una mejora de RAM en un futuro para así poder aprovechar el sistema al máximo.

La máquina host estará totalmente cerrada, solo se utilizará para acceder al entorno virtualizado, de esta forma podemos evitar que las máquinas virtuales se vean comprometidas a causa de una infección en la máquina host.

La estructura de máquinas será la siguiente:

3.1. Máquina 1. Ocio

Esta máquina, como su nombre indica, se utilizará para el ocio. Constará de un SO Windows 7 Professional SP1 con 5GB de RAM ya que los videojuegos requieren habitualmente una gran cantidad de memoria para poderse ejecutar correctamente. Además, se dispondrá de un disco duro de 400 GB ya que tanto videojuegos como películas ocupan una cantidad considerable de espacio en disco. Esta máquina tendrá un acceso completo a internet, es decir, no se realizará ningún tipo de operación en el firewall sobre esta máquina.

3.2. Máquina 2. Correo

Esta máquina se utilizará para la lectura de correo. Constará de un SO Xubuntu 16.04.1 LTS con 1 GB de RAM, aunque el mínimo necesario para el sistema operativo es 512 MB, y la lectura de correo no debería suponer una gran carga para la máquina, al disponer de 1GB nos aseguramos de que la máquina está sobrada de capacidad y no dará problemas como atascarse en tareas sencillas.

Finalmente, constará de un disco duro de 10 GB, ya que no va ser necesario almacenar información pesada en la máquina, pero podría ser interesante guardar algún archivo adjunto del correo. Los únicos puertos que tendrá accesibles serán los que permiten la recepción y envío de correo electrónico.

3.3. Máquina 3. Compras

Esta máquina se utilizará para realizar compras por internet. Constará de un SO Xubuntu 16.04.1 LTS con 1 GB de RAM. Al igual que en el caso anterior, a pesar de necesitar solo 512 MB para funcionar el SO, se prefiere disponer de más memoria RAM para evitar problemas de sobrecarga en tareas sencillas, además de que algunos navegadores consumen bastante memoria y es recomendable estar seguros de que la máquina irá holgada.

Sólo dispondrá de 10 GB de almacenamiento, ya que no va ser necesario almacenar información pesada en la máquina. En esta máquina solo se tendrá acceso al puerto 80, y si fuera necesario abrir algún puerto adicional para las pasarelas de pago. A pesar de las medidas de seguridad, en esta máquina es necesaria también la precaución del usuario para asegurarse de introducir siempre sus datos bancarios en sitios seguros.

3.4. Máquina 4. Trabajo

Esta máquina se utilizará sólo para la redacción de memorias y artículos, así pues, no tendrá acceso a internet, sin embargo, se podrán copiar a ella archivos a través de una carpeta compartida con las máquinas de correo electrónico y navegación, pues se puede dar el caso de pertenecer a un equipo de trabajo y necesitar compartir archivos por correo, o puede que se encuentre algún artículo interesante en el web en relación al trabajo que se esté realizando en el momento.

La máquina constará de un SO Ubuntu 16.04 LTS con 4 GB de RAM. Se utilizará una cantidad moderadamente grande ya que, además de redactar memorias, se podría utilizar para programación y para la realización de operaciones de cálculo relativamente grandes. Además, dispondrá de un disco duro de 100 GB, también una cantidad moderadamente grande, para evitar problemas de almacenamiento.

3.5. Máquina 5. Navegar

Esta máquina se utilizará para navegar por internet y acceder a las redes sociales. Aunque estas actividades podrían considerarse como ocio, y por tanto podrían realizarse en la primera máquina descrita, se ha preferido separar estas tareas en una máquina aparte por considerarse un vector de ataque considerable, además de tratarse de datos bastante sensibles, y sería muy fácil que un malware se colara descargando contenidos audiovisuales o visionándolos online desde páginas no seguras, o incluso desde páginas seguras, y este tuviera acceso a todos los datos personales de nuestros perfiles en redes sociales, lo cual puede suponer una serie de riesgos y pérdidas, sobre todo a nivel personal.

Constará de un SO Ubuntu 16.04 LTS con 2 GB de RAM, como se ha explicado en máquinas anteriores, a pesar de que este sistema solo necesita 1GB de RAM para funcionar correctamente, se prefiere asegurar que la máquina dispone de suficiente memoria y facilitar por tanto que el funcionamiento de la misma sea fluido. Además tendrá 100 GB de disco duro por si se desean guardar imágenes, fotos, etc. provenientes tanto de las redes sociales como de cámaras o móviles personales.

3.6. Descripción del firewall

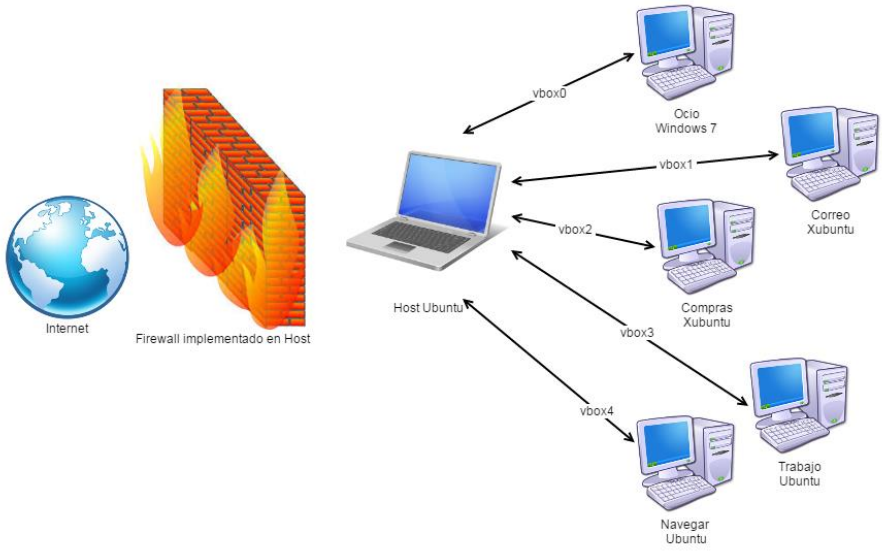


Figura 1: Esquema de la red

Las máquinas se conectan a internet mediante una conexión de tipo "Adaptador Sólo-Anfitrión". Este tipo de red en principio no ofrece salida a internet, pero al realizar el redireccionamiento NAT se podrá dar acceso a internet, pue se redirigirá todo el tráfico de cada interfaz a la interfaz wlp5s0.

La máquina host no podrá recibir ni enviar paquetes, simplemente servirá como firewall para el resto de máquinas.

En la máquina Ocio se tendrá acceso a todos los puertos, ya que se utilizará para jugar videojuegos, ver películas, etc.

En la máquina Correo solo se tendrá acceso a los puertos de correo electrónico, que serían 110/TCP y 995/TCP (Cifrado) para POP3, y 25/TCP,587/TCP (alternativo para clientes de correo) y 465/TCP (SMTPS) para SMTP.

En la máquina Compras habrá acceso a los puertos 80/TCP para HTTP y 443/TCP para HTTPS, pues requiere de conexión a internet.

En la máquina Trabajo no habrá acceso a internet de ninguna forma para evitar cualquier tipo de ataque posible.

Finalmente, en la máquina Navegar, al igual que la máquina Compras, tendrá abiertos los puertos 80/TCP para HTTP y 443/TCP para HTTPS, pues se utilizará mayoritariamente para conectarse a redes sociales.

4. Creación del entorno

Para crear el entorno seguro se ha seleccionado el hipervisor de tipo 2 VirtualBox. En este entorno se han creado las distintas máquinas virtuales que se han descrito en el apartado anterior. A continuación, se explicará paso a paso el proceso de creación de cada máquina.

Como inicialmente el proyecto se iba a realizar en un host Windows, se puede observar que las capturas de pantalla de la creación de las máquinas se encuentran en un entorno Windows, ya que al toparme con dificultades a la hora de realizar el proyecto en este entorno decidí exportar el servicio virtualizado y utilizar un host Linux.

4.1. Creación de las máquinas

Como las máquinas se crean básicamente de la misma forma solo se explicará la creación de una de ellas. También se explicará el proceso de clonado, ya que puede ser útil cuando se crean varias máquinas con las mismas características. La máquina con la que se va a explicar la creación es con la máquina Ocio, que es la más potente de todas.

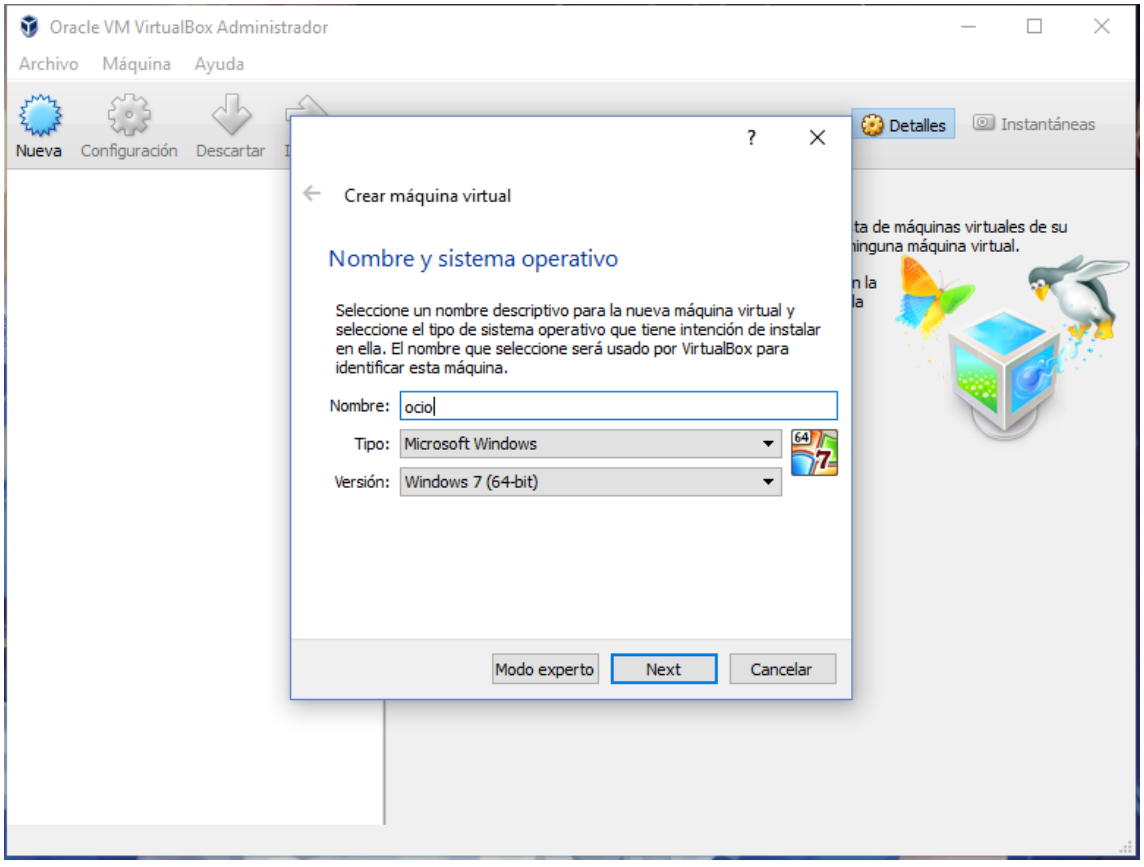


Figura 2: Creación de la máquina

Como se puede observar en la Figura 1, en este primer cuadro de diálogo se selecciona el nombre de la máquina, el tipo de sistema operativo, y la versión. En este caso se ha seleccionado Windows 7.

Tras pulsar el botón “Siguiente” nos aparece un nuevo cuadro de diálogo en el que se puede seleccionar la cantidad de memoria RAM de la que dispondrá la máquina. En este caso se le han otorgado 5 GB, ya que esta máquina está orientada a jugar a videojuegos, ver series, etc.

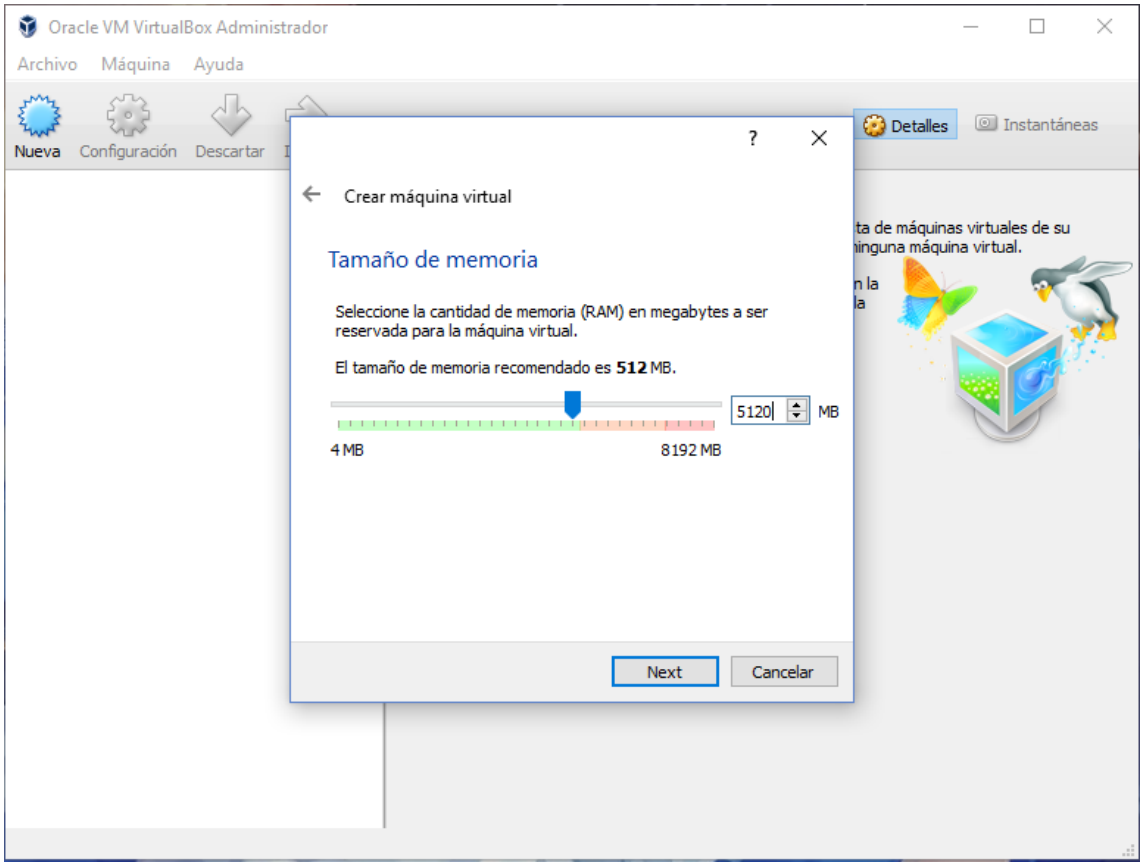


Figura 3: Asignación de memoria

Tras pulsar “Siguiente”, se abre un nuevo cuadro de diálogo en el que se ofrece la posibilidad de crear un nuevo disco duro para la máquina, no añadirle disco duro o reutilizar un archivo de disco duro virtual ya existente. Como se puede ver en la Figura 3, en este caso se escoge la opción de crear un nuevo disco duro para la máquina, esto se realizará igual en todas las máquinas virtuales.

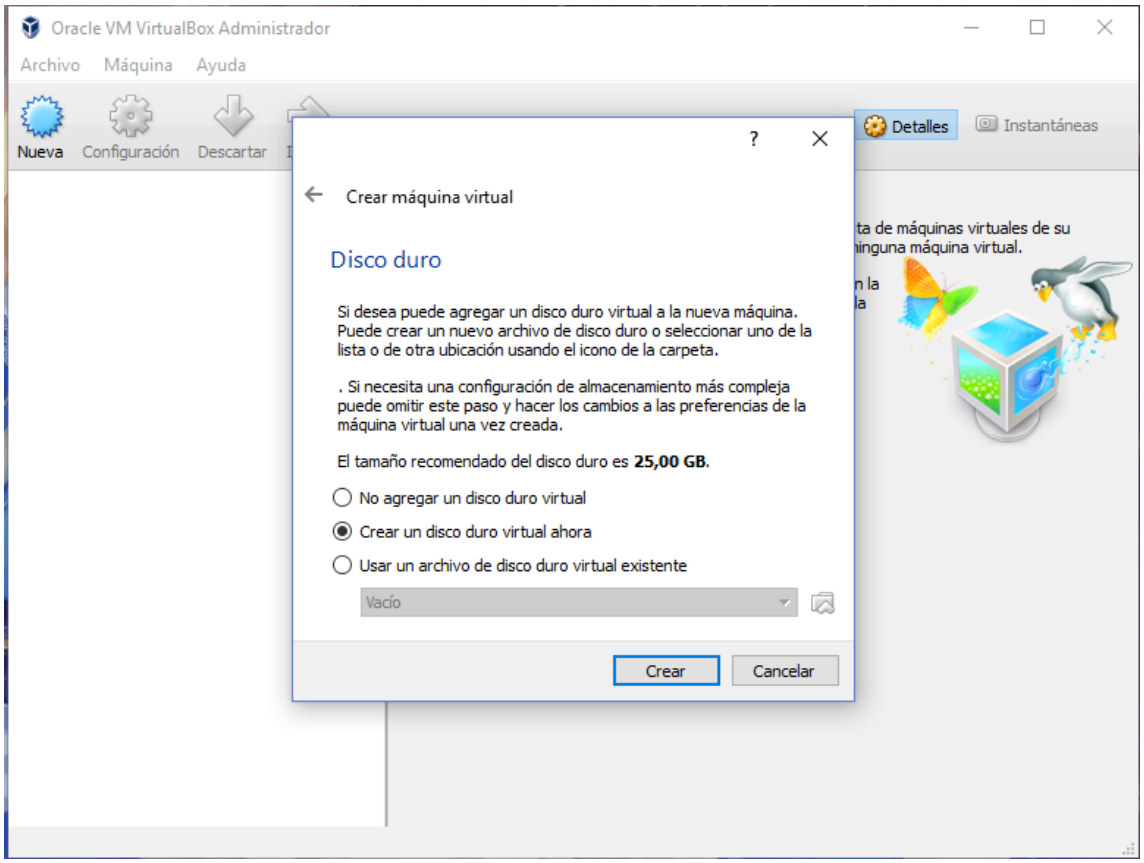


Figura 4: Creación del disco duro

Tras pulsar “Crear” se muestra un nuevo cuadro de diálogo en el que permite elegir el tipo de disco duro a crear, en este caso se selecciona la opción VDI VirtualBox Disk Image, como se puede observar en la Figura 4.

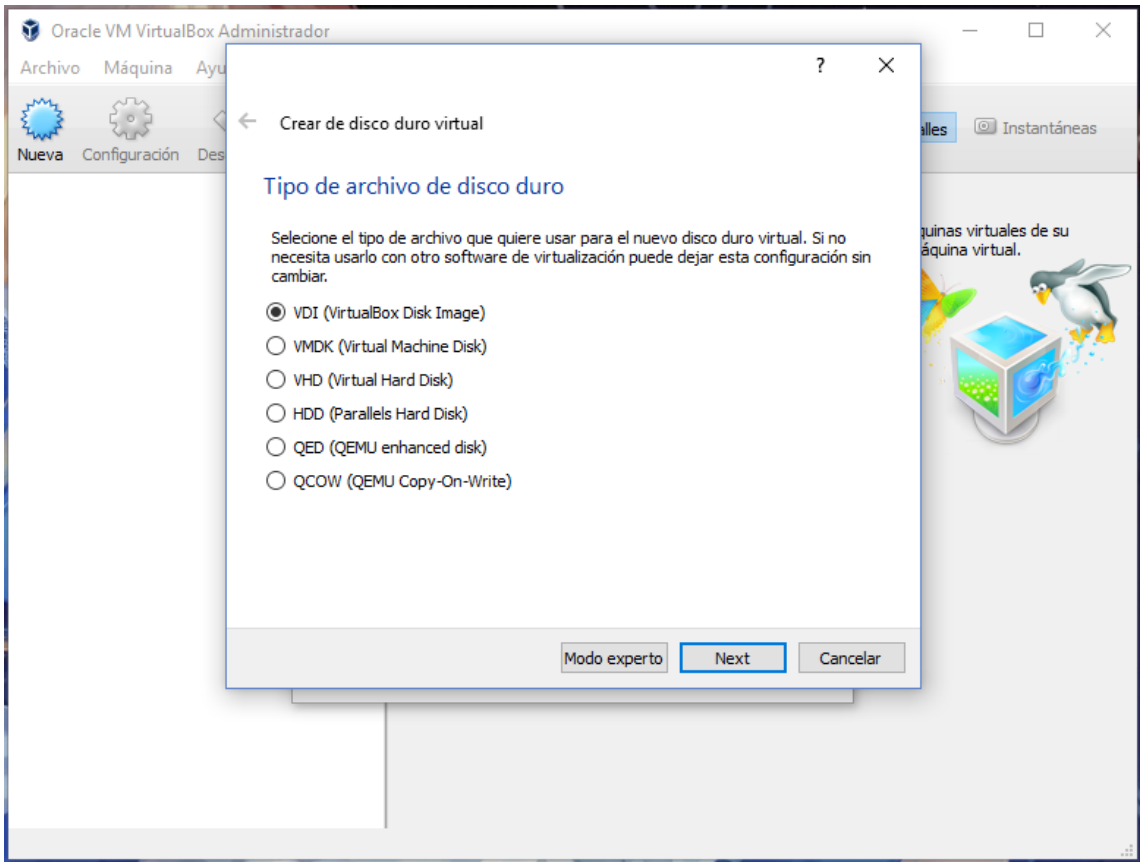


Figura 5: Selección del tipo de disco duro

Tras pulsar el botón “Siguiente”, se muestra un nuevo cuadro de diálogo que ofrece dos opciones de reserva para el espacio del disco duro, reserva dinámica y tamaño fijo. Se escoge la opción “Reservado dinámicamente”. Aunque en esta máquina se va a utilizar poco espacio de disco, en las otras máquinas donde el tamaño reservado será mayor, es más interesante tener el espacio reservado dinámicamente, pues en lugar de apartar el espacio, este se irá llenando conforme crezca el disco duro virtual hasta completar el tamaño asignado, aunque funcione más despacio. Esto se hará de la misma forma para todas las máquinas virtuales del entorno.

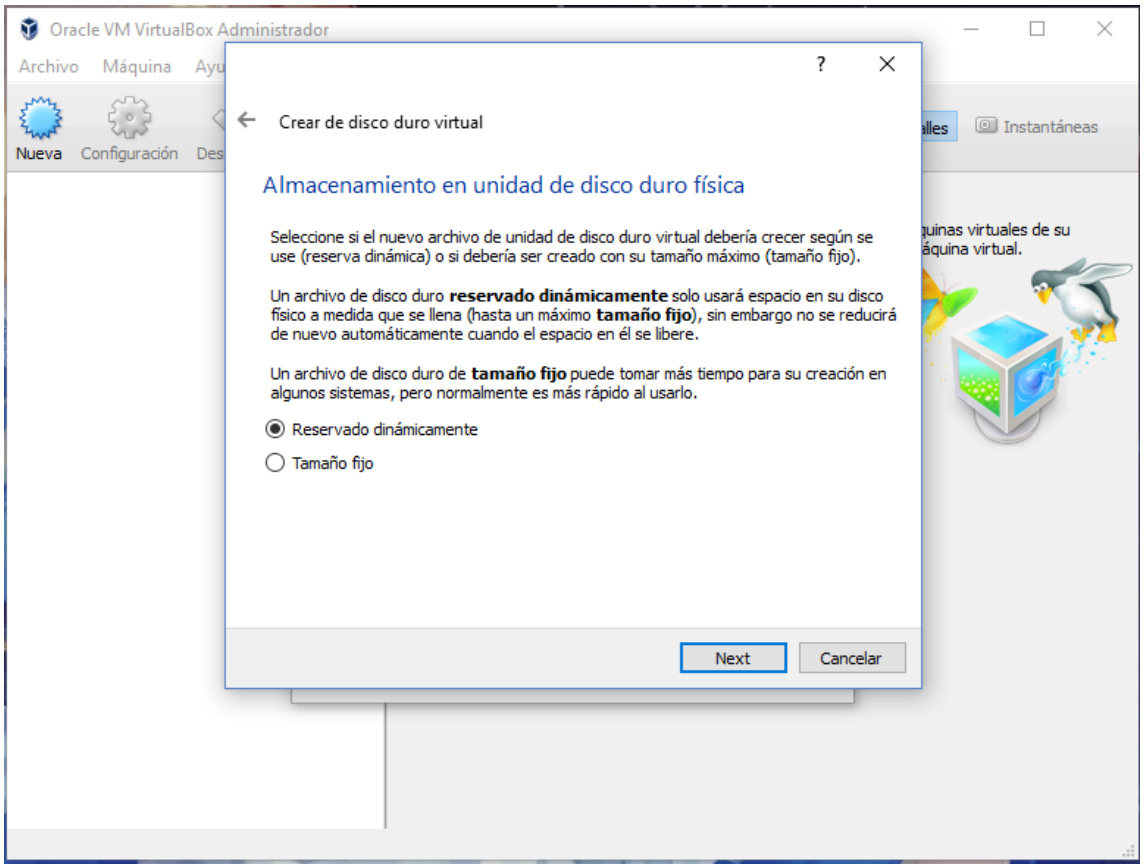


Figura 6: Elección de la reserva del disco

Finalmente, tras pulsar el botón “Siguiente” se muestra un último cuadro de diálogo en el que se selecciona el tamaño del disco duro virtual. Como esta máquina está orientada al ocio, se asignan 400 GB de disco duro.

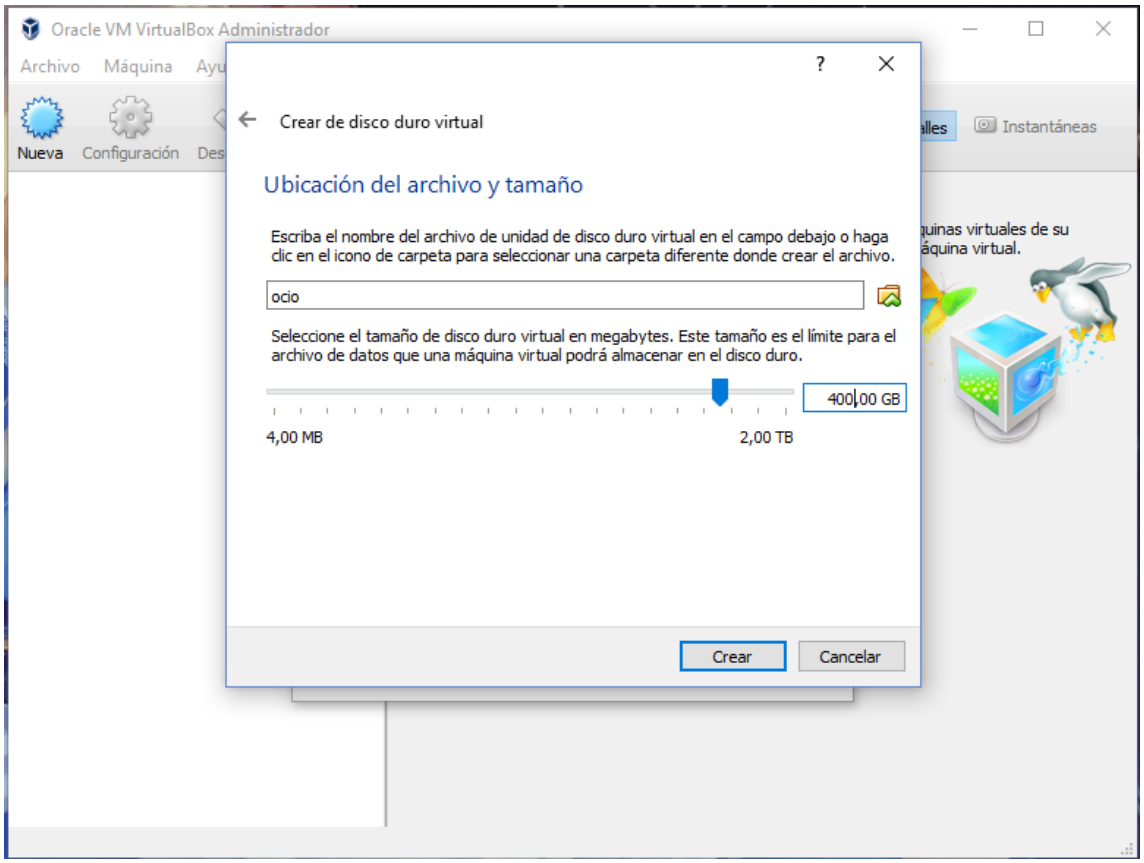


Figura 7: Elección del tamaño del disco

4.2. Creación de máquinas mediante clonado

La máquina Compras es idéntica en características a la máquina Correo, por este motivo en lugar de crear una nueva máquina virtual con idénticas características, se ha optado por utilizar la opción de clonación de máquinas de VirtualBox. Para ello, se posiciona el puntero sobre la máquina a clonar, en este caso Compras, y se hace clic derecho sobre ella. En el menú desplegable que aparece, se selecciona la opción “Clonar…” como se puede observar en la Figura 7.

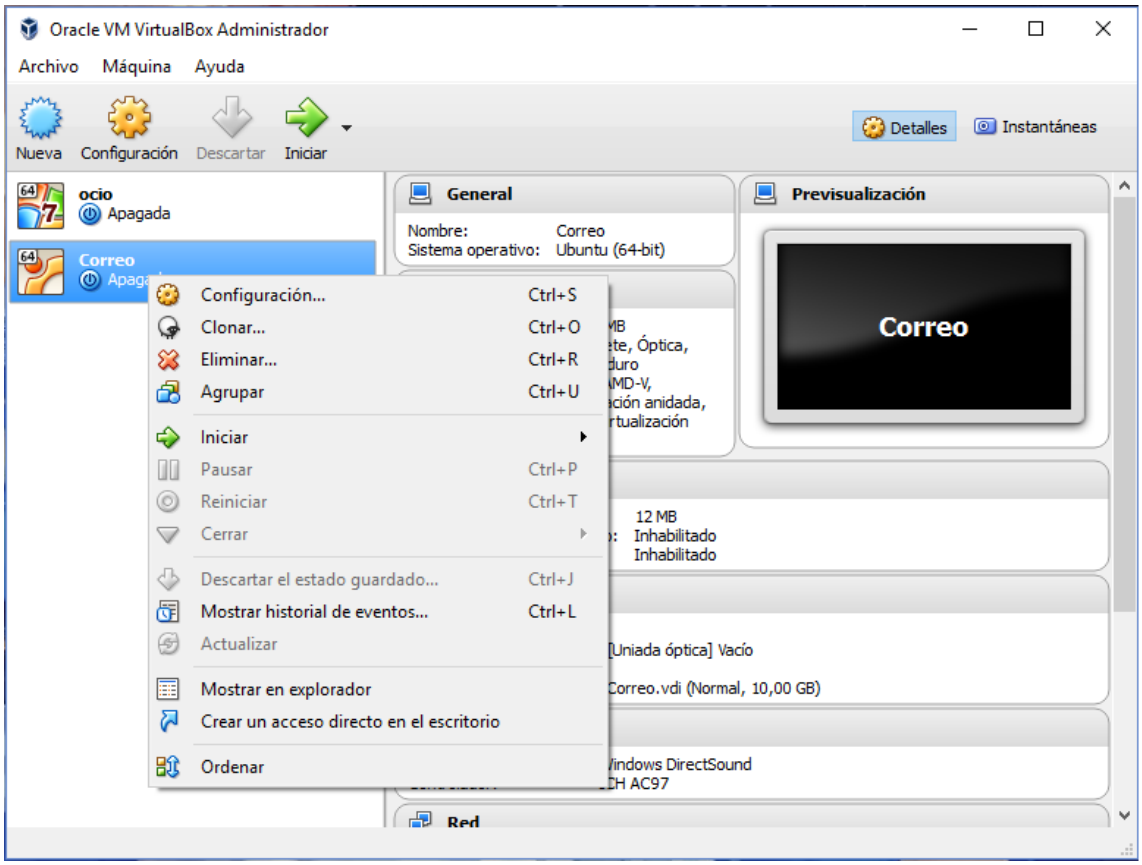


Figura 8: Creación de la máquina virtual mediante clonación

Tras pulsar sobre esta opción se abre un nuevo cuadro de diálogo en el que se solicita un nombre para la nueva máquina y se ofrece la posibilidad de reinicializar la dirección MAC de las tarjetas de red. Se marca esta opción para que se cree una nueva tarjeta de red virtual para la nueva máquina. De esta forma se evitan MACs duplicadas.

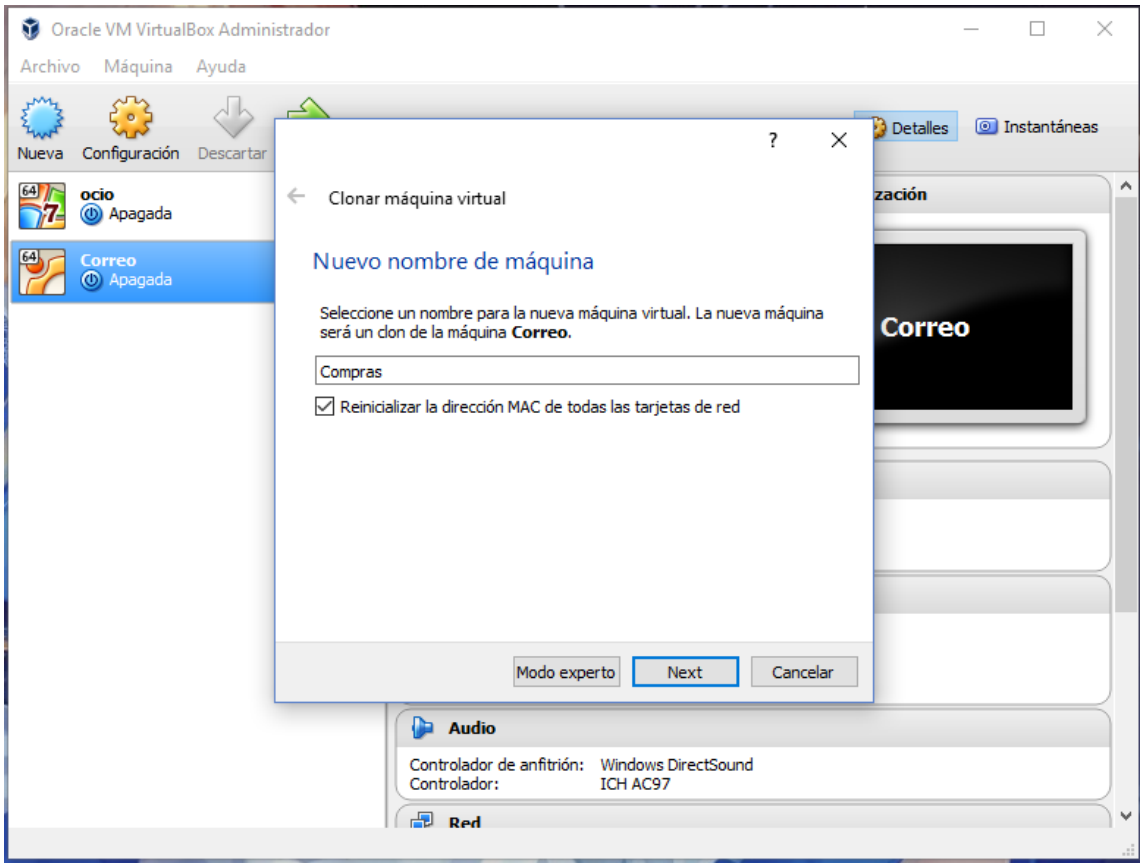


Figura 9: Seleccionar el nombre de la máquina virtual

Finalmente se selecciona la forma en que se desea clonar la máquina. VirtualBox ofrece dos estilos de clonación: completa y enlazada. En la clonación enlazada, la máquina clonada depende totalmente de la máquina original, si se deseara mover la máquina clonada a otro host, sería necesario mover también la máquina original, sin embargo, con la clonación completa se crea una copia exacta de la máquina origen. Se elige la opción de clonación completa para evitar problemas si en algún momento se deseara prescindir de la máquina Correo.

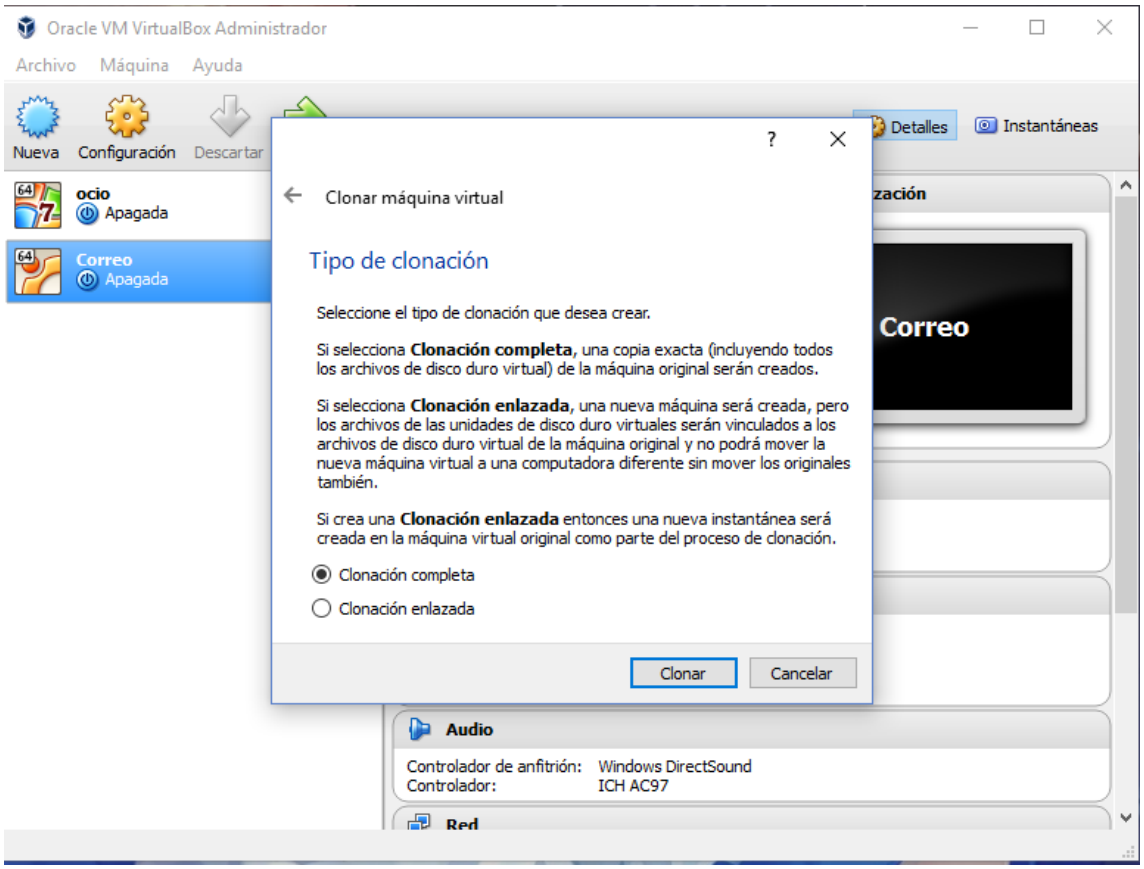


Figura 10: Selección del tipo de clonación

Tras pulsar el botón “Clonar”, el proceso finaliza y la máquina Compras está creada.

4.3. Creación y asignación de las interfaces de red

Para crear las redes de cada máquina como “Adaptador Sólo-Anfitrión” es necesario crear una serie de interfaces de red virtuales, esto se hace desde ‘Archivo’→ ‘Preferencias’ como se ve en las siguientes figuras:

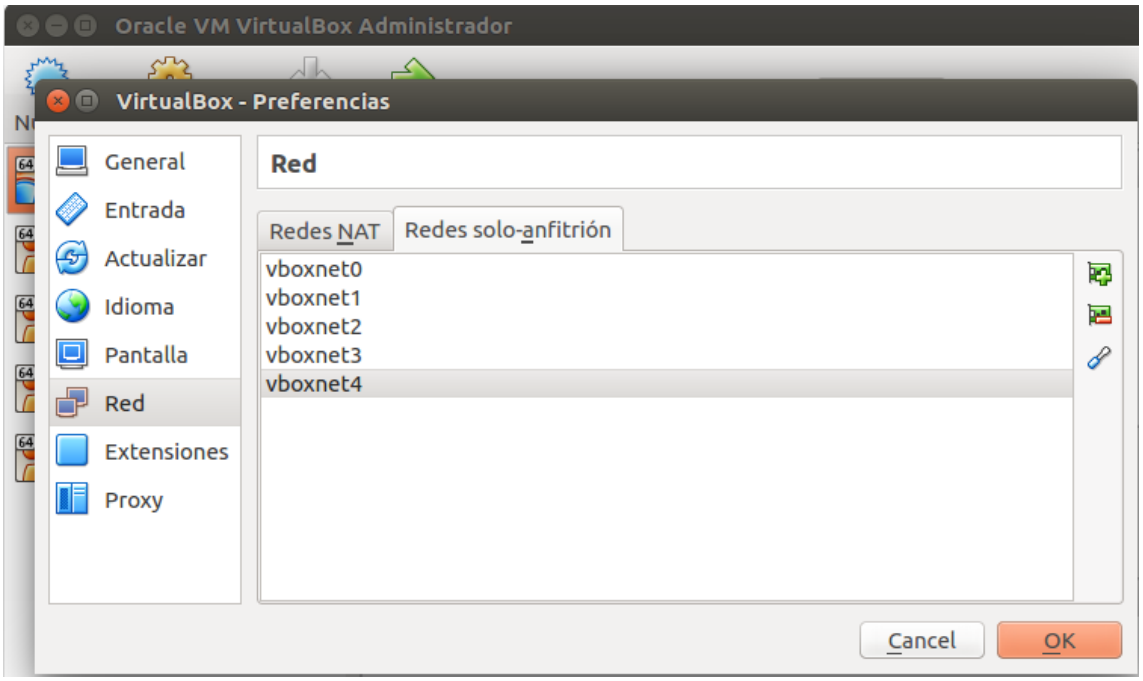


Figura 11: Creación de las redes

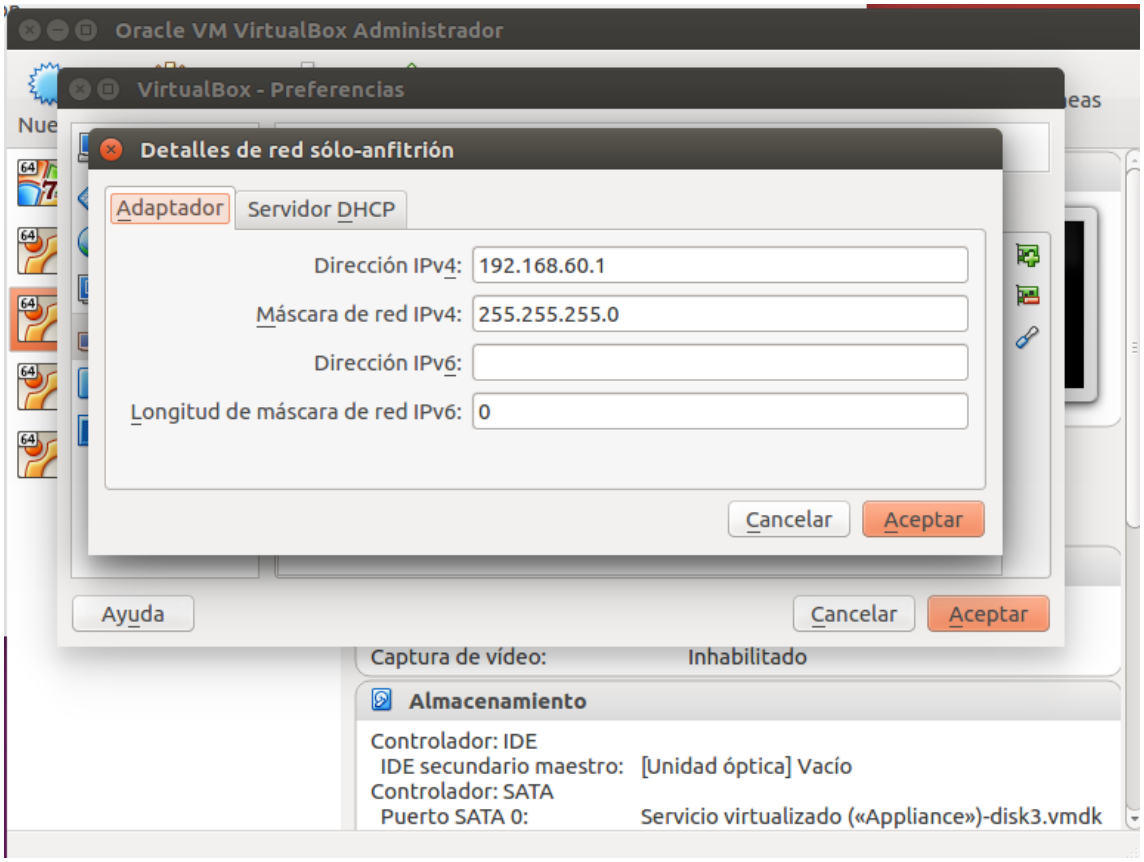


Figura 12: Asignación de las ip

Cada interfaz tendrá su correspondiente ip siendo vboxnet0 →192.168.56.1… vboxnet4 → 192.168.60.1

Después de crear las interfaces de red, se asignará una interfaz a cada máquina, quedando de esta forma:

- Ocio → vboxnet0
- Correo → vboxnet1
- Compras → vboxnet2
- Trabajo → vboxnet3
- Navegar → vboxnet4

Para añadir una interfaz a la máquina bastará con entrar en la configuración de la máquina, en el apartado 'Red', elegir como opción de conexión 'Adaptador sólo-anfitrión' y asignar una de las interfaces de red creadas anteriormente a la máquina:

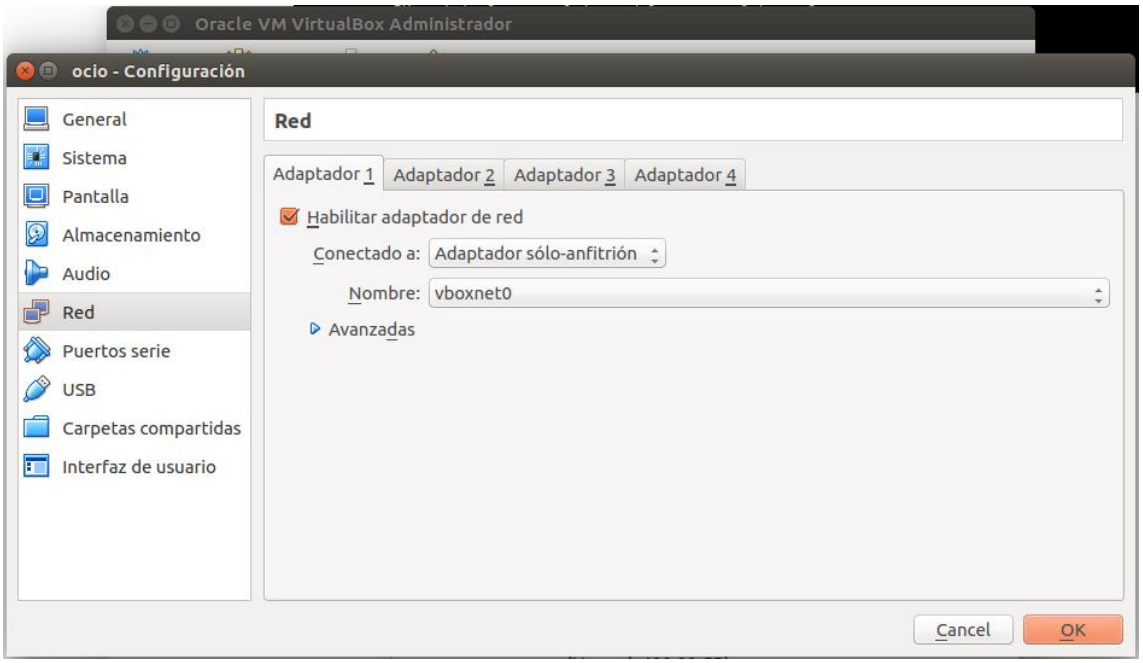


Figura 13: Asignación de la interfaz

Además, a cada máquina se le asignará una ip fija que tendrá como puerta de enlace la ip de su correspondiente interfaz, este proceso se puede ver en las siguientes figuras.

Para Windows: Para modificar la ip de una conexión, el primer paso es acceder a 'Panel de Control' → 'Redes e Internet' → 'Conexiones de red'. Se selecciona la red que se desea cambiar y se hace clic derecho, como se puede observar en la Figura 13:

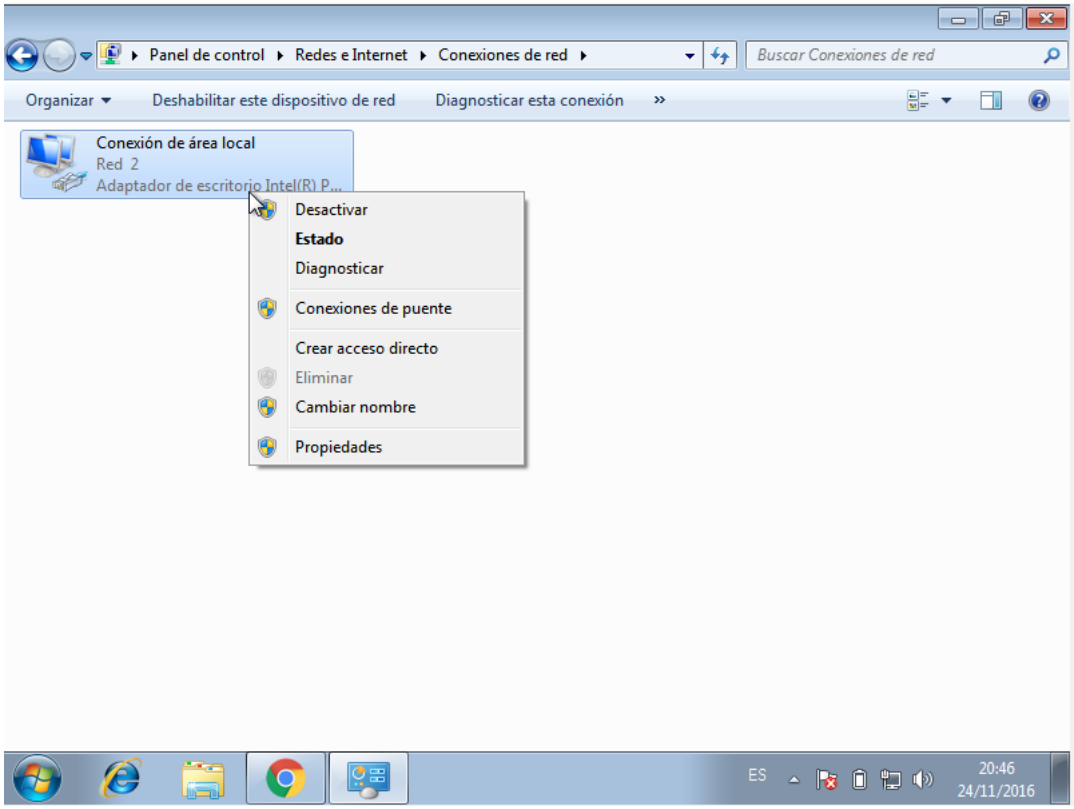


Figura 14: Asignación de la ip

A continuación, se pulsa la opción ‘Propiedades’, abriéndose así un diálogo dónde seleccionamos ‘Protocolo de Internet versión 4 (TCP/IPv4)’:

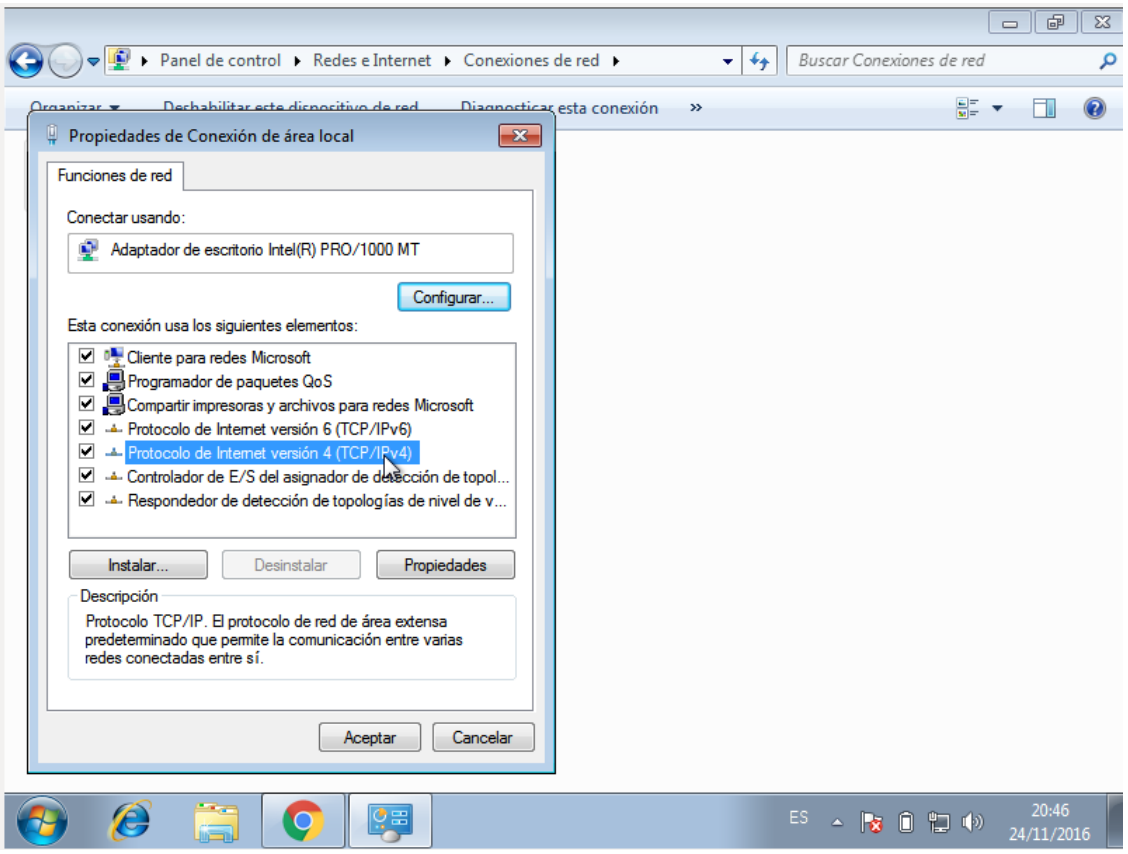


Figura 15: Selección del protocolo a modificar

Finalmente, en los ajustes del protocolo, marcamos la opción usar la siguiente dirección ip dónde indicamos como ip la dirección 192.168.56.100, y le indicamos que su puerta de enlace es la dirección ip que hemos puesto anteriormente como dirección de la interfaz de red vboxnet0: 192.168.56.1

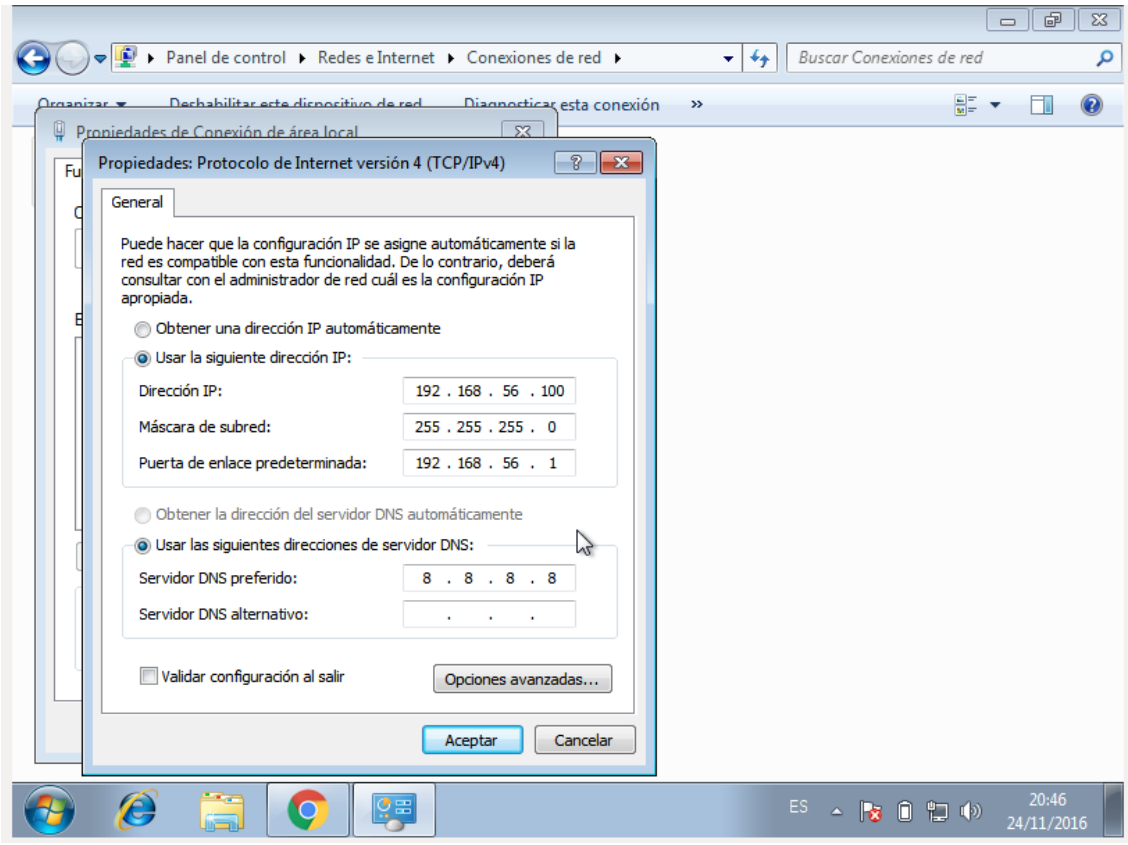


Figura 16: Asignación ip y puerta de enlace

Para Ubuntu y Kubuntu: Para modificar la dirección ip de una conexión, abrimos el panel de conexiones de Ubuntu/Kubuntu, seleccionamos nuestra conexión y pulsamos editar, apareciendo el siguiente diálogo dónde se asigna la MAC a la conexión:

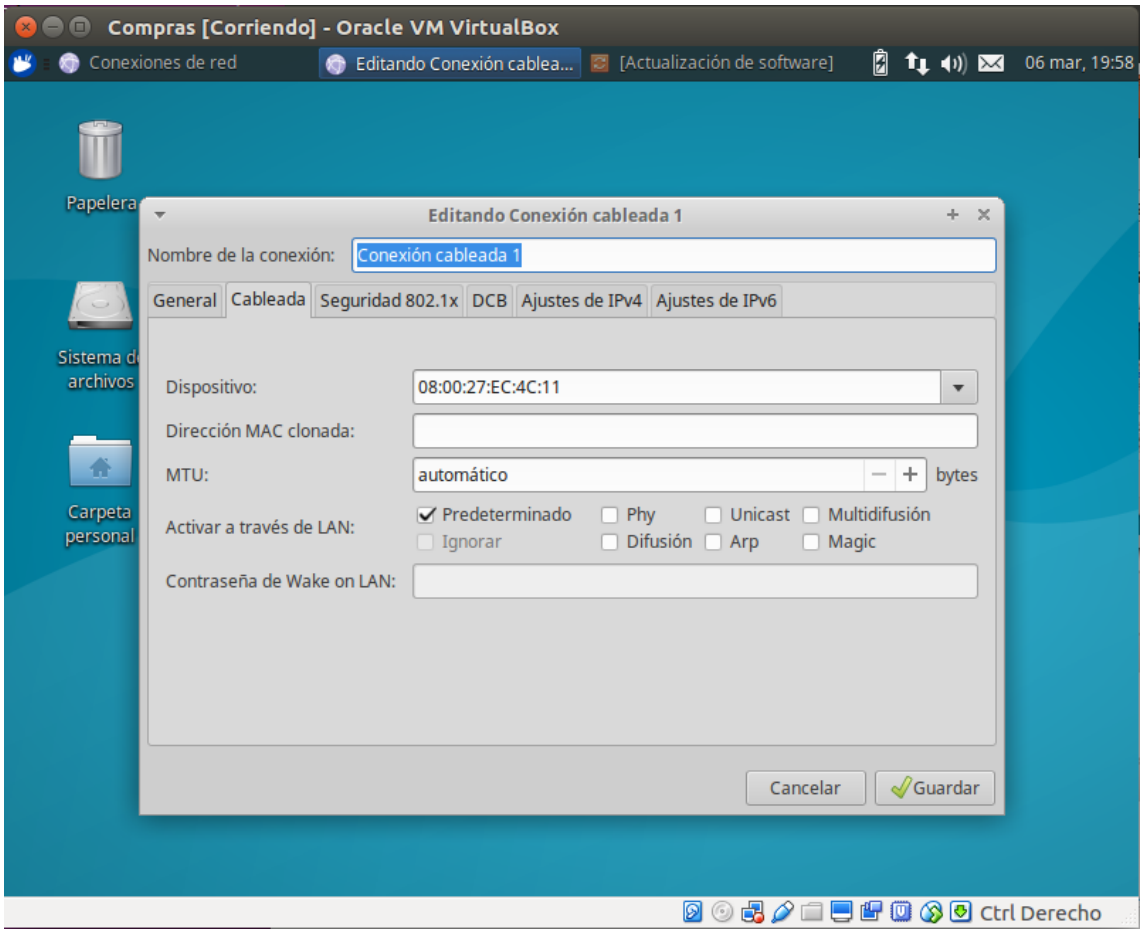


Figura 17: Asignación de la MAC

A continuación, nos dirigimos a la pestaña Ajustes de IPv4, el campo 'Método' tiene por defecto la opción de obtener la IP por DHCP, pero en este caso, se cambia el método a 'Manual' y añadimos una IP, al igual que en la máquina Windows, asignamos como ip la dirección: 192.168.58.100 y como puerta de enlace la dirección de la red: 192.168.58.1:

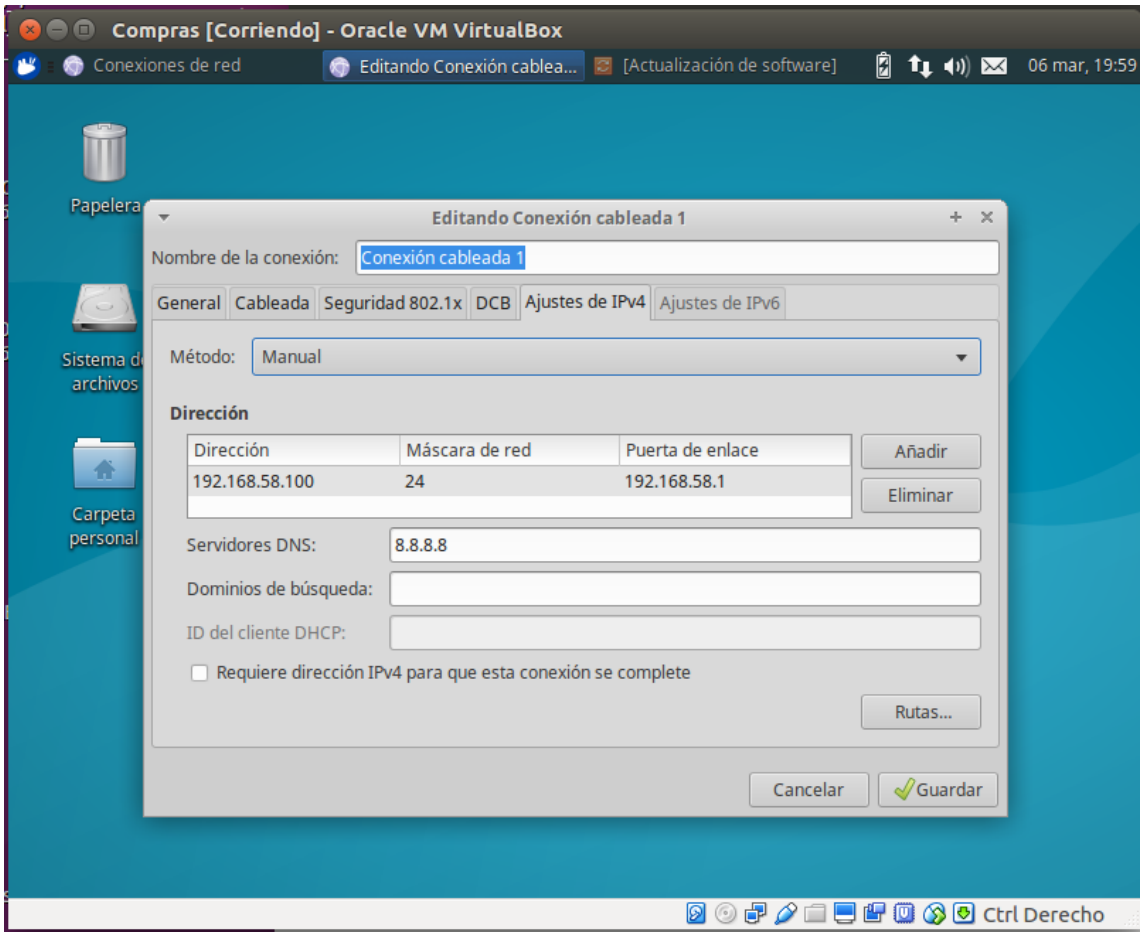


Figura 18: Asignación de la IP

5. Creación del Firewall

5.1. Concepto de firewall. Netfilter

Un firewall es un sistema de seguridad de red que monitoriza y controla el tráfico entrante y saliente en una red basándose en una serie de reglas. Generalmente un firewall establece una barrera entre una red interna segura y una red externa, como puede ser internet, que se asume como una red no segura. Los firewall normalmente se dividen en dos categorías: firewalls de red, o firewalls basados en host. Los firewalls de red filtran tráfico entre dos o más redes. Los firewall basados en host proporcionan un filtro de tráfico entrante y saliente de la máquina host. El firewall por defecto en Linux y el más utilizado es Netfilter.

Netfilter es una arquitectura de filtro de paquetes para los núcleos de Linux 2.4 y 2.6. El filtrado se hace en el mismo núcleo en las capas 2, 3 y 4 del modelo OSI, es decir, los vínculos dados, red y transporte. Por ejemplo, es capaz de actuar a bajo nivel en las interfaces ethernet, en la pila IP y en los protocolos de transporte como TCP. El filtrado no tiene estado: como Netfilter sólo inspecciona los encabezamientos de los paquetes, es muy veloz y no conlleva tiempo de espera.

Se pueden inspeccionar los contenidos de los paquetes (protocolos aplicativos) usando extensiones, pero este trabajo se delega a herramientas de usuario. Dicho de otro modo, netfilter es un firewall que actúa a nivel del protocolo. El programa usuario que permite actuar sobre las reglas de filtrado es iptables.

5.2. Iptables

Iptables es una aplicación que le permite a un administrador de sistema configurar las tablas, cadenas y reglas del firewall. Debido a que iptables requiere privilegios elevados para operar, el único que puede ejecutarlo es el superusuario.

El formato del comando iptables es el siguiente: iptables [-t <table-name>] <command> <chain-name> <parameter-1> \<option-1> <parameter-n> <option-n>.

La opción <table-name> permite al usuario seleccionar una tabla diferente a la tabla predeterminada a usar con el comando. La opción <command> indica una acción específica a realizar, tal como anexar o eliminar la regla especificada por la opción <chain-name>. Luego de la opción <chain-name> se encuentran un par de parámetros y opciones que definen qué pasará cuando un paquete coincide con la regla.

La estructura de un comando iptables puede variar en longitud y complejidad en función de su propósito. Por ejemplo, mientras que un comando para borrar una regla de una cadena puede ser muy corto, un comando diseñado para filtrar

paquetes de una subred particular usando un conjunto de parámetros específicos y opciones puede ser mucho más largo.

Iptables ofrece gran cantidad de comandos para añadir reglas de filtrado de paquetes, etc. En este apartado se explicarán los comandos que se utilizarán para crear nuestro firewall, además de algún comando que resulte interesante para el uso de iptables:

- -A Añade la regla iptables al final de la cadena especificada. Este es el comando utilizado para simplemente añadir una regla cuando el orden de las reglas en la cadena no importa.
- -D Borra una regla de una cadena en particular por número (como el 5 para la quinta regla de una cadena). Puede también teclear la regla entera e iptables borrará la regla en la cadena que corresponda.
- -F Libera la cadena seleccionada, que borra cada regla de la cadena. Si no se especifica ninguna cadena, este comando libera cada regla de cada cadena.
- -h Proporciona una lista de estructuras de comandos, así como también un resumen rápido de parámetros de comandos y opciones.
- -I Inserta una regla en una cadena en un punto especificado por un valor entero definido por el usuario. Si no se especifica ningún número, iptables colocará el comando en el tope de la cadena.
- -L Lista todas las reglas de la cadena especificada tras el comando. Para ver una lista de todas las reglas en todas las cadenas en la tabla por defecto filter, no se debe especificar ninguna cadena o tabla. De lo contrario, la sintaxis siguiente deberá utilizarse para listar las reglas en una cadena específica en una tabla en particular: iptables –L <chain-name> -t <table-name>
- -P Configura la política por defecto para una cadena en particular, de tal forma que, cuando los paquetes atraviesen la cadena completa sin cumplir ninguna regla, serán enviados a un objetivo en particular, como puedan ser ACCEPT o DROP.
- -X Borra una cadena especificada por el usuario. No se permite borrar ninguna de las cadenas predefinidas para cualquier tabla.
- -Z Pone ceros en los contadores de byte y de paquete en todas las cadenas de una tabla en particular.

Dentro de iptables hay 5 tablas, que son zonas en las que cada cadena se puede aplicar:

- raw: filtra los paquetes antes que cualquier otra tabla. Se utiliza principalmente para configurar exenciones de seguimiento de conexiones en combinación con el target NOTRACK.
- filter: es la tabla por defecto a no ser que se use la opción –t vista anteriormente.

- nat: se utiliza para la traducción de dirección de red (por ejemplo, el redirección de puertos). Debido a las limitaciones en iptables, el filtrado no se debe hacer aquí.
- mangle: se utiliza para la alteración de los paquetes de red especializados.
- security: se utiliza para reglas de conexión de red Mandatory Access Control.

De estas 5 tablas, en nuestro firewall solo se utilizarán dos: la tabla por defecto, filter, y la tabla nat.

Además, dentro de cada tabla, hay cadenas, las cuales son listas de reglas que ordenan los paquetes de red, la tabla filter contiene tres cadenas integradas:

- Todo el tráfico entrante, dirigido a la máquina, se hace pasar a través de la cadena INPUT.
- Todo el tráfico saliente, generado localmente, pasa a través de la cadena OUTPUT.
- Todo el tráfico enrutado, que no se ha suministrado localmente, pasa a través de la cadena FORWARD.

La tabla nat contiene las siguientes cadenas integradas, además de INPUT y OUTPUT, explicadas previamente:

- PREROUTING para alterar paquetes cuando entran a través del router.
- POSTROUTING para alterar paquetes que están a punto de salir por el router.

Una vez que especificados ciertos comandos iptables, incluyendo aquellos para añadir, anexar, eliminar, insertar o reemplazar reglas dentro de una cadena, las tablas que existen y los tipos de cadena, se requieren parámetros para construir una regla de filtrado de paquetes.

- -d Configura el nombre de la máquina destino, dirección IP o red de un paquete que coincide con la regla. Cuando se coincida una red, se soportan los siguientes formatos de direcciones IP o máscaras de red:
    - N.N.N.N/M.M.M.M — Donde N.N.N.N es el rango de direcciones IP y M.M.M.M es la máscara de la red.
    - N.N.N.N/M — Donde N.N.N.N es el rango de direcciones IP y M es la máscara de bit.
- -i Configura la interfaz de red entrante, tal como eth0 o ppp0.
  Con iptables, este parámetro opcional puede ser usado solamente con las cadenas INPUT y FORWARD cuando es usado con la tabla filter y la cadena PREROUTING con las tablas nat y mangle.

  Este parámetro también soporta las siguientes opciones especiales:

    - El carácter de exclamación: ! invierte la directriz, es decir, se excluye de esta regla cualquier interfaz especificada.

    - El carácter de suma (+) es un carácter tipo comodín utilizado para hacer coincidir todas las interfaces con una cadena de caracteres especificada. Por ejemplo, el parámetro -i eth+ aplicará esta regla a cualquier interfaz ethernet pero excluirá cualquier otra interfaz, tal como, ppp0.
    - Si el parámetro -i se utiliza sin especificar ninguna interfaz, todas las interfaces estarán afectadas por la regla.
- -j Salta a un objetivo particular cuando un paquete coincide con una regla particular. Los objetivos válidos a usar después de la opción -j incluyen las opciones estándar (ACCEPT, DROP, QUEUE y RETURN) así como también las opciones extendidas que están disponibles a través de los módulos cargados por defecto con el paquete RPM de Red Hat Enterprise Linux iptables, como LOG, MARK y REJECT, entre otros.
  Puede también dirigir un paquete coincidiendo esta regla a una cadena definida por el usuario fuera de la cadena actual, para aplicar otras reglas al paquete.
  Si no especifica ningún objetivo, el paquete pasa la regla sin llevar a cabo ninguna acción. A pesar de todo, el contador para esta regla se sigue incrementando en uno.
- -o Configura la interfaz de red de salida para una regla y puede ser usada solamente con las cadenas OUTPUT y FORWARD en la tabla de filtro y la cadena POSTROUTING en las tablas nat y mangle. Estos parámetros de opciones son los mismos que aquellos de la interfaz de entrada (-i).
- -p Configura el protocolo IP para la regla, el cual puede ser icmp, tcp, udp, o all, para coincidir todos los protocolos soportados. Además, se puede usar cualquier protocolo listado en /etc/protocols. Si esta opción es omitida cuando se esté creando una regla, la opción all es la opción por defecto.
- -s Configura la fuente para un paquete particular usando la misma sintaxis que el parámetro (-d).

Una vez que un paquete ha coincidido con una regla, la regla puede dirigir el paquete a un número de objetivos diferentes que deciden que se va a hacer con ese paquete. Cada cadena tiene un objetivo por defecto, el cual es usado si ninguna de las reglas en esa cadena coinciden con un paquete, o si ninguna de las reglas que coinciden con el paquete especifica un objetivo.

Los objetivos estándar son los siguientes:

- <user-defined-chain> Reemplazando <user-defined-chain> con el nombre de una cadena definida por el usuario dentro de la tabla, este objetivo pasa el paquete a la cadena objetivo.
- ACCEPT Permite que el paquete se mueva hacia su destino (o hacia otra cadena, si no ha sido configurado ningún destino para seguir a esta cadena).

- DROP — Deja caer el paquete sin responder al solicitante. El sistema que envía el paquete no es notificado de este fallo.
- QUEUE — El paquete se pone en una cola para ser manejado por una aplicación en el espacio de usuario.
- RETURN — Para la verificación del paquete contra las reglas de la cadena actual. Si el paquete con un destino RETURN cumple una regla de una cadena llamada desde otra cadena, el paquete es devuelto a la primera cadena para retomar la verificación de la regla allí donde se dejó. Si la regla RETURN se utiliza en una cadena predefinida, y el paquete no puede moverse hacia la cadena anterior, el objetivo por defecto de la cadena actual decide qué acción llevar a cabo.

Además de estos objetivos estándar, se pueden usar otros más con extensiones llamadas módulos de objetivos (target modules). Existen varios módulos extendidos de objetivos, la mayoría de los cuales tan sólo se aplican a tablas o situaciones específicas. Un par de estos módulos, de los más populares son:

- LOG Registra todos los paquetes que coinciden esta regla. Puesto que los paquetes son registrados por el kernel, el archivo /etc/syslog.conf determina dónde se escribirán estas entradas de registro. Por defecto, se colocan en el archivo /var/log/messages.

- REJECT Envía un paquete de error de vuelta al sistema remoto y deja caer el paquete. El objetivo REJECT acepta --reject-with <tipo> (donde <tipo> es el tipo de rechazo) el cual permite devolver información más detallada con el paquete de error. El mensaje port-unreachable es el tipo de error por defecto si no se usa otra opción.

5.3. Creación del firewall

Para la creación del firewall se ha creado un script de shell (.sh) el cual contiene las reglas a aplicar en cada máquina. Estas reglas indican qué hacer con cada paquete, y cómo hacer el redireccionamiento vía NAT.

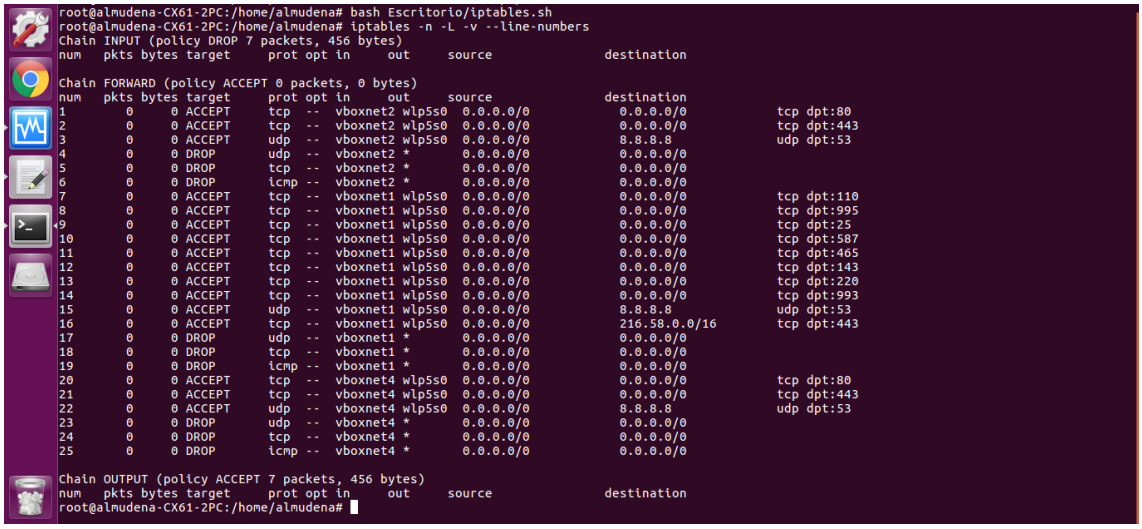
```
#!/bin/bash

IP_HOST="$(ifconfig | grep -A 1 'wlp5s0' | tail -1 | cut -d ':' -f 2 | cut -d ' ' -f 1)"

iptables -t nat -F
iptables -F

iptables -P INPUT DROP

#################################
#                               #
#   Máquina Compras             #
#                               #
#################################

iptables -A FORWARD -i vboxnet2 -o wlp5s0 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i vboxnet2 -o wlp5s0 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i vboxnet2 -o wlp5s0 -p udp --dport 53 -d 8.8.8.8  -j ACCEPT

iptables -A FORWARD -i vboxnet2 -p UDP -j DROP
iptables -A FORWARD -i vboxnet2 -p tcp -j DROP
iptables -A FORWARD -i vboxnet2 -p icmp -j DROP

iptables -t nat -A POSTROUTING -s 192.168.58.100 -j SNAT --to ${IP_HOST}

#################################
#                               #
#   Máquina Correo              #
#                               #
#################################

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 110 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 995 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 25 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 587 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 465 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 143 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 220 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 993 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p udp --dport 53 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 443 -d 216.58.0.0/16 -j ACCEPT

iptables -A FORWARD -i vboxnet1 -p udp -j DROP
iptables -A FORWARD -i vboxnet1 -p tcp -j DROP
iptables -A FORWARD -i vboxnet1 -p icmp -j DROP
```

Figura 19 Script Firewall (parte 1)

```
iptables -t nat -A POSTROUTING -s 192.168.57.100 -j SNAT --to ${IP_HOST}

#################################
#                               #
#   Máquina Navegar             #
#                               #
#################################

iptables -A FORWARD -i vboxnet4 -o wlp5s0 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i vboxnet4 -o wlp5s0 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i vboxnet4 -o wlp5s0 -p udp --dport 53 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -i vboxnet4 -p UDP -j DROP
iptables -A FORWARD -i vboxnet4 -p tcp -j DROP
iptables -A FORWARD -i vboxnet4 -p icmp -j DROP

iptables -t nat -A POSTROUTING -s 192.168.60.100 -j SNAT --to ${IP_HOST}

#################################
#                               #
#   Máquina Ocio                #
#                               #
#################################

iptables -t nat -A POSTROUTING -s 192.168.56.100 -j SNAT --to ${IP_HOST}
```

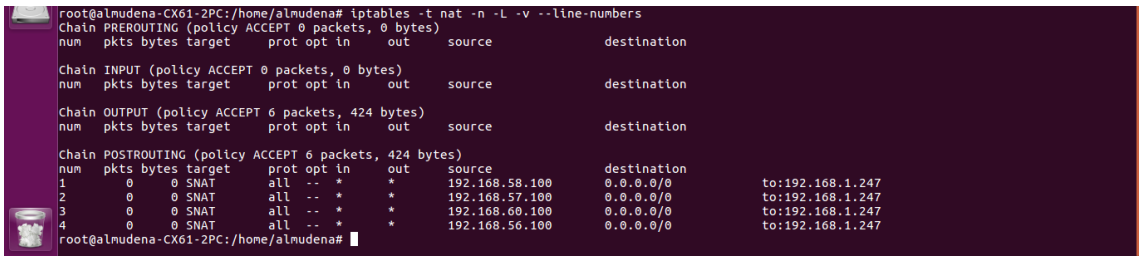
Figura 20: Script Firewall (parte 2)

Tras ejecutar el script, las tablas quedan de la siguiente forma:



```
root@almudena-CX61-2PC:/home/almudena# bash Escritorio/iptables.sh
root@almudena-CX61-2PC:/home/almudena# iptables -n -L -v --line-numbers
Chain INPUT (policy DROP 7 packets, 456 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     tcp  --  vboxnet2 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:80
2        0     0 ACCEPT     tcp  --  vboxnet2 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:443
3        0     0 ACCEPT     udp  --  vboxnet2 wlp5s0  0.0.0.0/0            8.8.8.8             udp dpt:53
4        0     0 DROP       udp  --  vboxnet2 *       0.0.0.0/0            0.0.0.0/0
5        0     0 DROP       tcp  --  vboxnet2 *       0.0.0.0/0            0.0.0.0/0
6        0     0 DROP       icmp --  vboxnet2 *       0.0.0.0/0            0.0.0.0/0
7        0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:110
8        0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:995
9        0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:25
10       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:587
11       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:465
12       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:143
13       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:220
14       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:993
15       0     0 ACCEPT     udp  --  vboxnet1 wlp5s0  0.0.0.0/0            8.8.8.8             udp dpt:53
16       0     0 ACCEPT     tcp  --  vboxnet1 wlp5s0  0.0.0.0/0            216.58.0.0/16       tcp dpt:443
17       0     0 DROP       udp  --  vboxnet1 *       0.0.0.0/0            0.0.0.0/0
18       0     0 DROP       tcp  --  vboxnet1 *       0.0.0.0/0            0.0.0.0/0
19       0     0 DROP       icmp --  vboxnet1 *       0.0.0.0/0            0.0.0.0/0
20       0     0 ACCEPT     tcp  --  vboxnet4 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:80
21       0     0 ACCEPT     tcp  --  vboxnet4 wlp5s0  0.0.0.0/0            0.0.0.0/0           tcp dpt:443
22       0     0 ACCEPT     udp  --  vboxnet4 wlp5s0  0.0.0.0/0            8.8.8.8             udp dpt:53
23       0     0 DROP       udp  --  vboxnet4 *       0.0.0.0/0            0.0.0.0/0
24       0     0 DROP       tcp  --  vboxnet4 *       0.0.0.0/0            0.0.0.0/0
25       0     0 DROP       icmp --  vboxnet4 *       0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT 7 packets, 456 bytes)
num   pkts bytes target     prot opt in     out     source               destination
root@almudena-CX61-2PC:/home/almudena#
```

Figura 21: Tabla filter



```
root@almudena-CX61-2PC:/home/almudena# iptables -t nat -n -L -v --line-numbers
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 6 packets, 424 bytes)
num   pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 6 packets, 424 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 SNAT       all  --  *      *       192.168.58.100       0.0.0.0/0           to:192.168.1.247
2        0     0 SNAT       all  --  *      *       192.168.57.100       0.0.0.0/0           to:192.168.1.247
3        0     0 SNAT       all  --  *      *       192.168.60.100       0.0.0.0/0           to:192.168.1.247
4        0     0 SNAT       all  --  *      *       192.168.56.100       0.0.0.0/0           to:192.168.1.247
root@almudena-CX61-2PC:/home/almudena#
```

Figura 22: Tabla nat

Tras la aplicación del firewall, las máquinas virtuales quedan blindadas para realizar su función. A continuación, se muestran una serie de pruebas de ejecución:

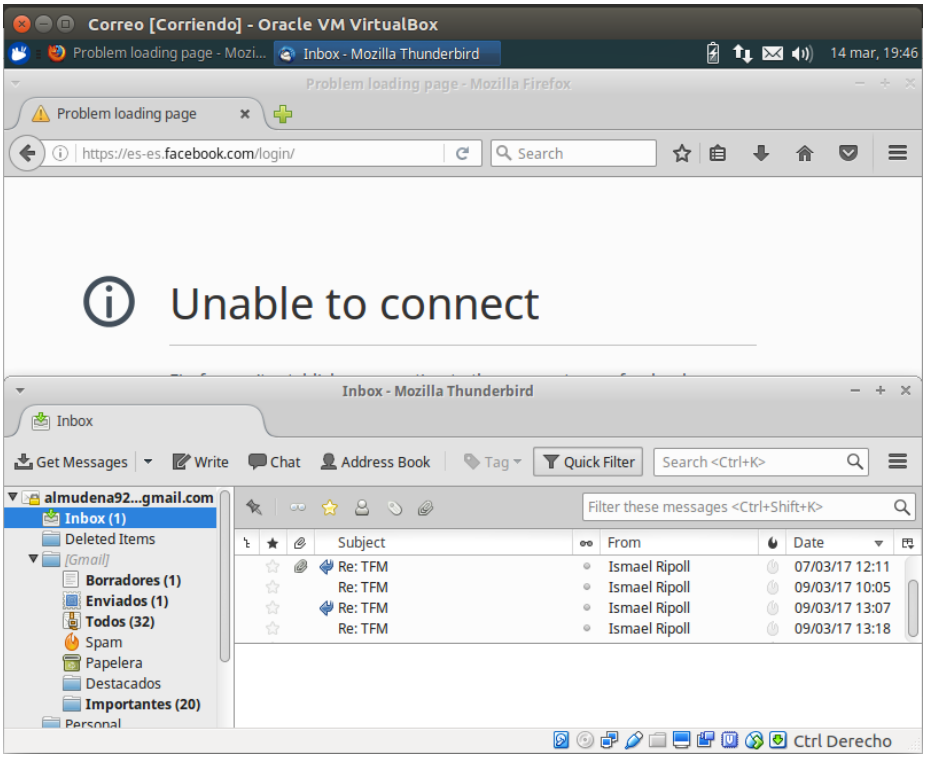


Figura 23: Máquina correo

Como se puede observar en esta máquina, no hay acceso a internet, pero si a la aplicación de correo electrónico.

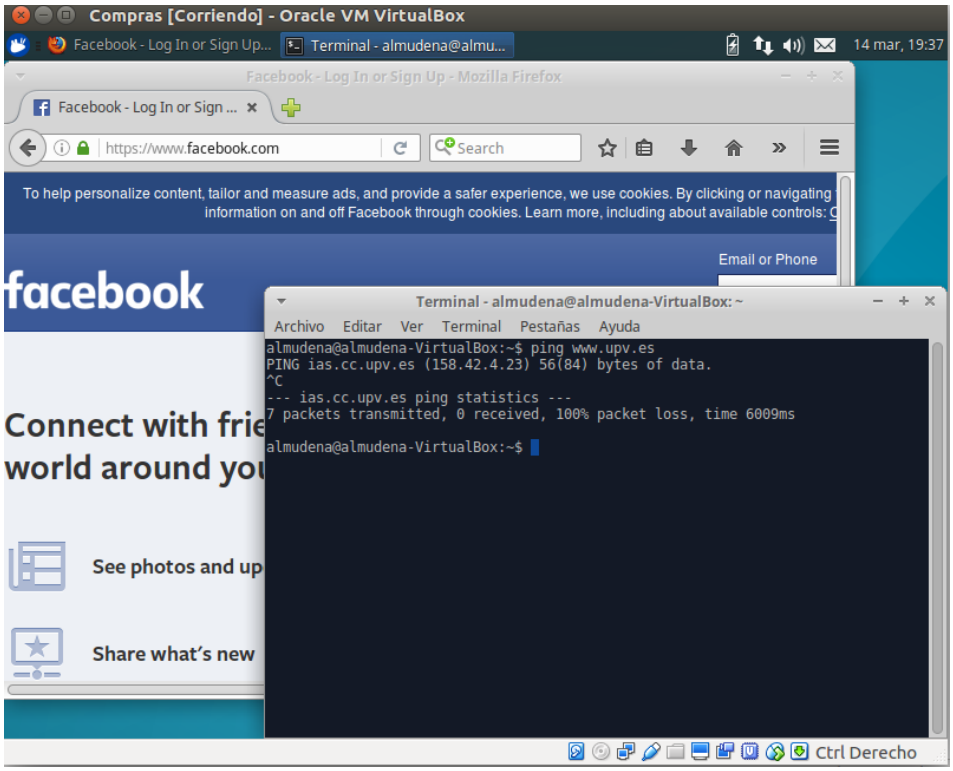


Figura 24: Máquina Compras

En la máquina compras, sin embargo, hay total acceso a internet, pero como se puede comprobar, no se puede hacer ping a otras páginas, pues el protocolo ICMP está cerrado.

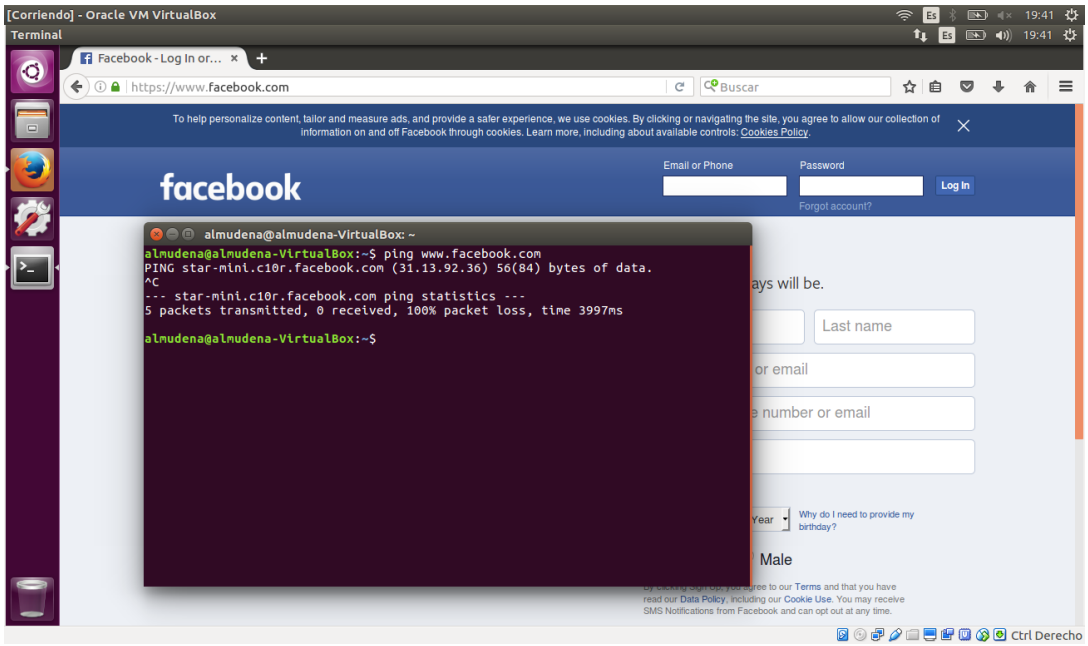


Figura 25: Máquina Navegar

En esta otra máquina, ocurre al igual que en la de Compras, pues sus reglas son idénticas.

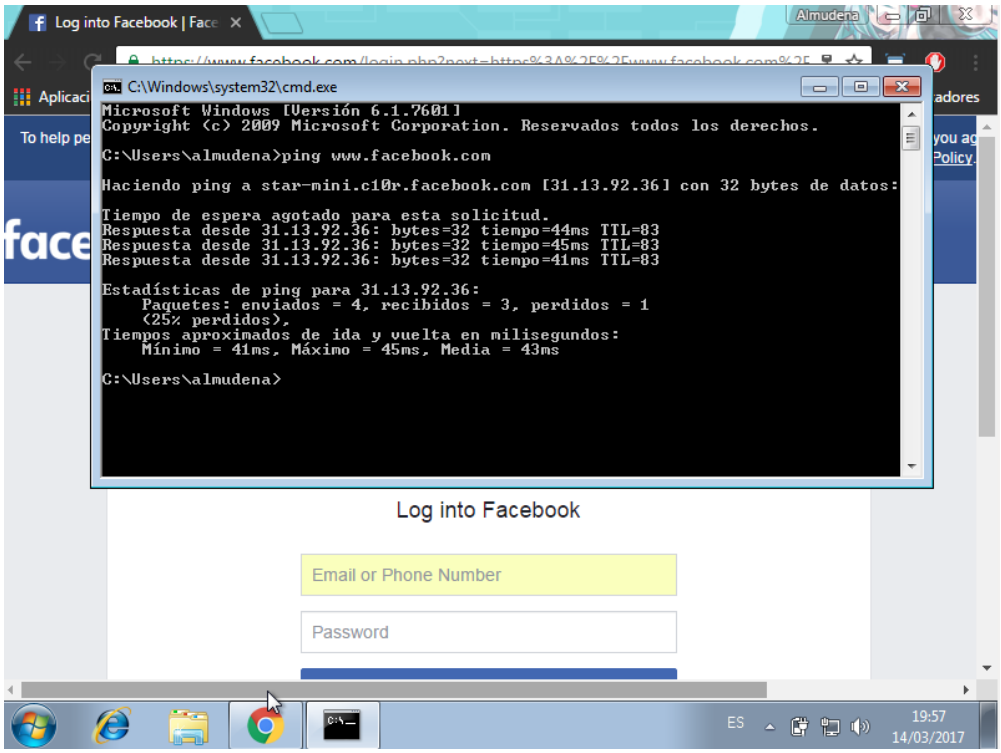


Figura 26: Máquina Ocio

En la máquina Ocio, solo se hace redirección NAT, así pues, tiene acceso completo a internet, es la menos segura de las máquinas. Se comprueba que se puede acceder a internet y hacer ping.

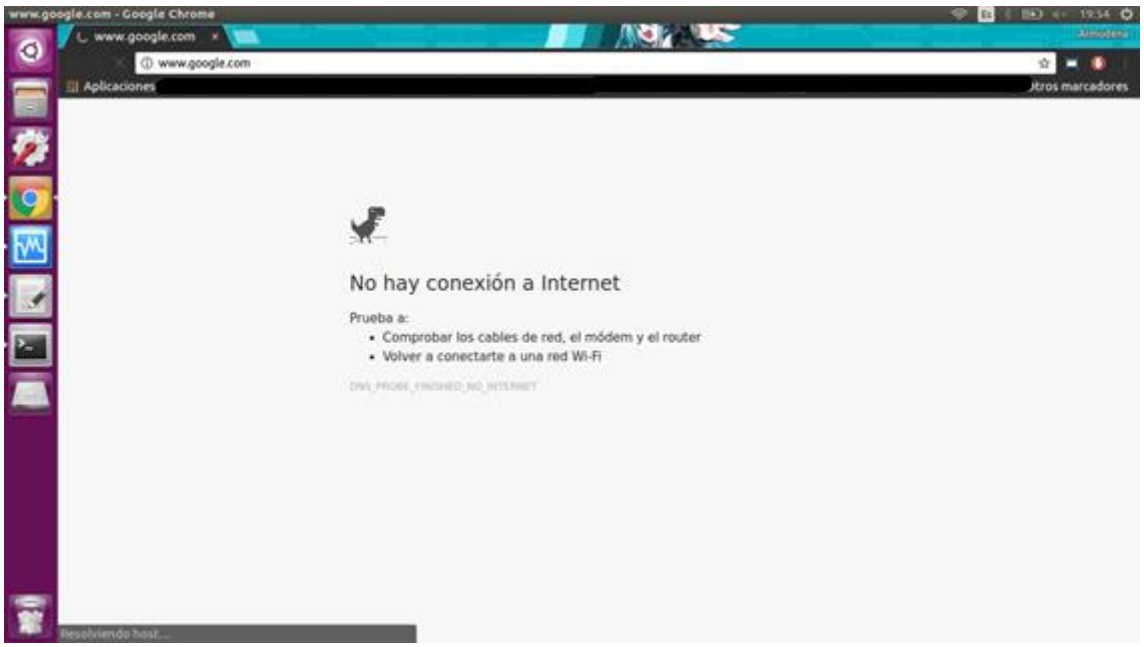


Figura 27: Máquina Host

En la máquina host, al rechazar todos los paquetes entrantes, no se puede establecer conexión.

5.4. Discusión de la solución

En este apartado se procederá a explicar una instrucción tipo de cada una de las instrucciones utilizadas en el firewall, ya que algunas de las reglas se utilizan múltiples veces.

IP_HOST="$(ifconfig | grep -A 1 'wlp5s0' | tail -1 | cut -d ':' -f 2 | cut -d ' ' -f 1)"

Con esta instrucción, creamos una variable y le asignamos la ip de la interfaz wlp5s0.

iptables -t nat -F

iptables -F

iptables -P INPUT DROP

Estas tres primeras reglas son las que aparecen al principio del script, las dos primeras sirven para borrar todo lo que hubiera en las tablas de nat y en la tabla principal, con la tercera regle indicamos que, por defecto, se desecharán todos los paquetes de la cadena INPUT, esto permite que nuestro host no tenga conexión a internet.

La más utilizada, con variantes respecto a los puertos, es la siguiente:

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 110 -j ACCEPT

El comando –A, se utiliza para añadir reglas, y FORWARD indica que la regla se añadirá a la cadena FORWARD. –i indica cuál es la interfaz de entrada, que en nuestro caso, siempre será la interfaz de la máquina virtual, mientras que –o indica la interfaz de salida, la cual es la wlp5s0, que es nuestra interfaz de wlan. Con –p indicamos el protocolo que queremos filtrar, tcp en este caso y con --dport el puerto por el cual que se filtra. Finalmente, con –j indicamos el objetivo al que dirigir la regla, y con ACCEPT indicamos que los paquetes que coincidan con la regla deben ser aceptados. Con esta regla se indica que se deben dejar pasar los paquetes que salgan por el puerto 110 desde la interfaz vboxnet1

Para poder habilitar la conexión con el dns, se ha utilizado la siguiente regla:

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p udp --dport 53 -d 8.8.8.8 -j ACCEPT

Lo único que cambia respecto a la regla anterior, es que, en este caso, el protocolo filtrado es el protocolo udp y el puerto 53, que corresponden a las peticiones para dns. Además, se ha añadido –d 8.8.8.8, esto indica que la ip destino es la 8.8.8.8, dns de Google, con esta regla se indica que se deben aceptar los paquetes con destino 8.8.8.8 que salgan por el puerto 53 udp desde la interfaz vboxnet1.

En la máquina de correo, se ha tenido que añadir una regla un poco más compleja, ya que a la hora de conectar nunca llegaba a realizar la conexión. Utilizando el comando netstat -putan, se ha podido observar que había algunos paquetes intentando salir por el puerto 443 que quedaban en estado SYN-SENT, ya que este puerto se encontraba cerrado en la máquina originalmente. Esto se debe a que thunderbird, el cliente de correo de mozilla, realiza una comprobación de certificados a través de este puerto, que corresponde a https. Debido a este motivo, se ha añadido un filtrado un poco más completo que el de otras reglas

iptables -A FORWARD -i vboxnet1 -o wlp5s0 -p tcp --dport 443 -d 216.58.0.0/16 -j ACCEPT

En este caso, además de añadir una dirección, al igual que en el caso del dns, se ha añadido una máscara, y se hace un filtrado en base a los 16 primeros bits de la dirección, puesto que cada vez cambia la ip.

Al final de la sección del firewall para cada máquina, se han añadido las siguientes tres líneas:

iptables -A FORWARD -i vboxnet1 -p udp -j DROP

iptables -A FORWARD -i vboxnet1 -p tcp -j DROP

iptables -A FORWARD -i vboxnet1 -p icmp -j DROP

Con éstas, indicamos que todos los paquetes pertenecientes al protocolo udp, tcp o icmp, que no hayan sido aceptados ya por una regla anterior, se desecharán.

Finalmente, en todas las secciones se añade una regla en la tabla nat, que es la que permite que las máquinas virtuales tengan conexión a internet:

iptables -t nat -A POSTROUTING -s 192.168.57.100 -j SNAT --to ${IP_HOST}

-t nat indica que esta regla se añade a la tabla nat y con POSTROUTING indicamos a que cadena se va a añadir. –s indica la ip que se va a redireccionar, en este caso se trata de la ip interna de la máquina, la de su interfaz de red, que no es la ip de la vboxnet, con SNAT se indica que se va a cambiar la dirección origen de los paquetes, justo antes de ser enviados, como indica la cadena POSTROUTING, finalmente, con la opción --to indicamos a que ip se va a determinar como ip origen, es decir, la dirección por la que vamos a cambiar nuestra ip.

6. Conclusiones

Como se ha podido observar en la discusión de la solución para poder montar un sistema como el explicado a lo largo de este documento, se requiere tener un conocimiento bastante amplio de redireccionamiento NAT, de iptables y tener muy clara la topología de la red para poder saber cómo se deben encaminar los paquetes, en qué momento se debe hacer la sustitución de la ip, cuál interfaz es la origen y cuál la destino, etc.

Como ya se ha mencionado en la introducción, en lugar de VMware, se ha decidido usar VirtualBox, ya que es un hipervisor bien conocido por mí, con las facilidades que eso suponía a la hora de trabajar, pues ya conozco la forma de crear las máquinas, qué tipos de redes permite crear y cómo crearlas, etc.

Para ayudarnos con la problemática de los puertos, es muy práctico el uso del comando netstat, con el cual se puede ver que puertos están en uso. Este programa se ha utilizado, por ejemplo, para resolver un problema a la hora de afinar el firewall de la máquina de Correo, como se ha podido ver en el apartado anterior.

Netfilter, concretamente iptables, como opción para crear un firewall es una herramienta muy potente, pues, pese a que en este proyecto se haya utilizado de forma muy básica, enfocándonos sobre todo en puertos y protocolos, iptables también ofrece la posibilidad de realizar filtrados más concretos, por ejemplo, filtrado por aplicación o por ip, el cual se ha visto al dar permisos para acceder al DNS. Una posibilidad de mejora del proyecto, sería utilizar la opción de filtrado por aplicación para, en la máquina de compras permitir acceso únicamente a páginas seguras de compras, como pueden ser Amazon, Aliexpress, eBay, etc.

7. Bibliografía

Qubes OS. (2016). *Qubes OS*. 1 Julio 2016, https://www.qubes-os.org/

Firewalls FAQ. (2016). *Faqs.org*. 10 Julio 2016, http://www.faqs.org/faqs/firewalls-faq/

Evolution of the Firewall Industry. (2016). *Docstore.mik.ua*. 5 Agosto 2016, http://docstore.mik.ua/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm

INGHAM, K., & FORREST, S. (2016). *A History and Survey of Network Firewalls*. 15 Agosto 2016, http://www.cs.unm.edu/~treport/tr/02-12/firewall.pdf

Red Hat Enterprise Linux 4: Manual de referencia. (2016). *http://web.mit.edu*. Retrieved 20 Noviembre 2016, http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-iptables-options.html

Altadill Izura, P. (2016). *IPTABLES manual práctico, tutorial de iptables con ejemplos*. 30 Noviembre 2016, http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf

netfilter/iptables project homepage - The netfilter.org project. (2016). *Netfilter.org*. 5 Diciembre 2016, from http://www.netfilter.org/

Chapter 6. Virtual networking. (2016). *Virtualbox.org*. 5 Diciembre 2016, https://www.virtualbox.org/manual/ch06.html

Ellingwood, J. (2017). *How To Forward Ports through a Linux Gateway with Iptables | DigitalOcean*. *Digitalocean.com*. 15 Febrero 2017, https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables

Rohaut, S. (2015). *LINUX* (3ª ed., p. Capítulo 8). [Cornellà de Llobregat]: ENI.