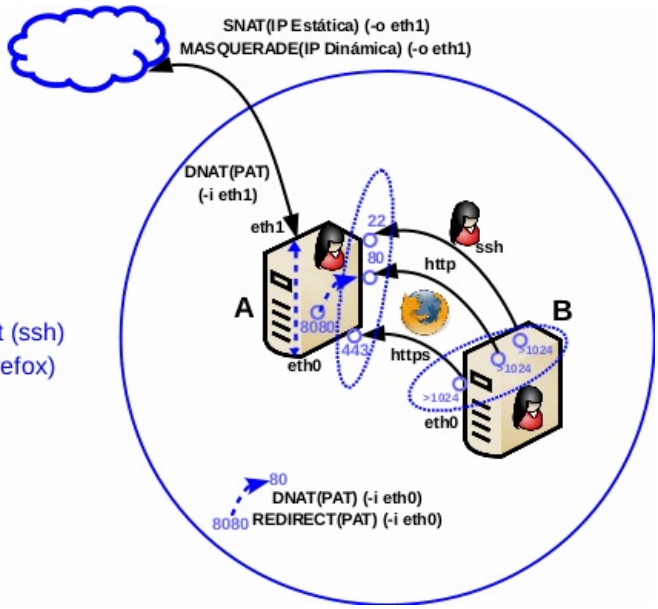
Firewall: iptables

ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0/24

Máquina virtual A:	**Máquina virtual B:**
Rede Interna(eth0) e NAT(eth1)	Rede Interna(eth0)
Servidor SSH: openssh-server	Cliente SSH: openssh-client (ssh)
Servidor Web: Apache (apache2)	Cliente Web: Navegador (firefox)
Firewall: iptables	
Router: eth0 — ip forward — eth1	
ISO: Kali Live amd64	ISO: Kali Live amd64
IP/MS: 192.168.120.100/24	IP/MS: 192.168.120.101/24

BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB



LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

NOTAS:

- Prerrequisitos:
  - **Servizo Web: Apache**
  - **Cifrado asimétrico: Certificado Apache**

- Firewall GNU/Linux: Paquete **iptables** (# apt update && apt -y install iptables && man iptables).

  Proxecto: **netfilter**

  Documentación de interese:
  - **https://github.com/frznlogic/iptables-tutorial**
  - **https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html#TRAVERSINGOFTABLES**

    ```
    $ git clone https://github.com/frznlogic/iptables-tutorial.git && less README
    ```

  - **https://wiki.archlinux.org/index.php/Iptables_(Espa%C3%B1ol)#Conceptos_b%C3%A1sicos**

- Servidor Web Apache: Paquete **apache2** (# apt update && apt -y install apache2).
- Cliente ssh GNU/Linux: Paquete **openssh-client (comando ssh)** (# apt update && apt -y install openssh-client).
- Servidor SSH GNU/Linux: Paquete **openssh-server** (# apt update && apt -y install openssh-server).
  Ficheiro de configuración: **/etc/ssh/sshd_config (man sshd_config)**

NOTAS:

- **Documentación oficial sobre netfilter/iptables**
- Prerrequisito: É necesario un kernel ≥2.4 ($ uname -r)
- iptables: Funciona a través de táboas, cadeas e regras.

nftables ➜ evolución de iptables

  - Existen unhas táboas creadas predeterminadas.
  - Pódense definir novas táboas, as cales tamén se poden eliminar.
  - As táboas conteñen cadeas e posúen unha política por defecto, sendo esta **ACCEPT**, é dicir, aceptar todos os paquetes.
  - As cadeas posúen as regras definidas no firewall, as cales permitirán por exemplo: aceptar(**ACCEPT**) e/ou impedir(**DROP**) paquetes.
  - Os paquetes poden ser identificados polo seu estado de conexión, ip orixe, ip destino, rede, porto orixe, porto destino...
  - Podemos aumentar as posibilidades de identificación de paquetes mediante a opción **-m** (iptables-extensions), por exemplo identificar paquetes por mac-address
  - iptables está activo tal que calquera paquete con orixe, destino hacia o host que posúe iptables atenderá as regras existentes neste.
  - **IMPORTANTE:** As regras nas cadeas lense secuencialmente de arriba hacia abaixo. Unha vez un paquete coincida cunha regra afectaralle esta e deixará a cadea do firewall, non lle afectando ningunha regra máis de iptables nesa cadea.

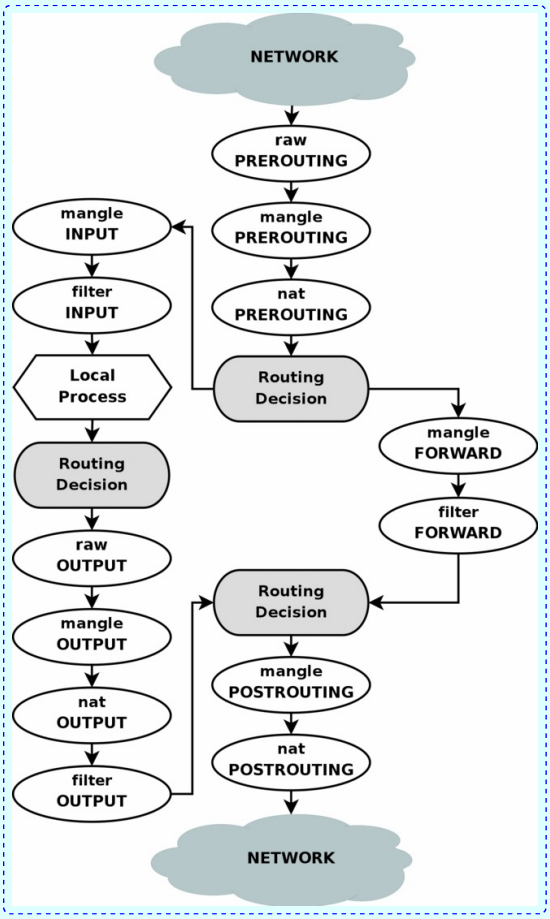


Fig. Tables Reverse - www.frozen.tux.net

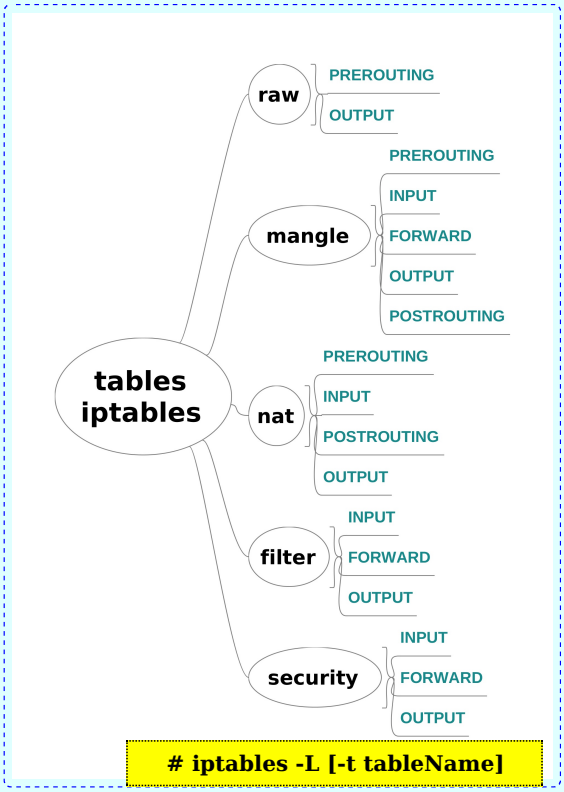


Fig. Táboas e cadeas

iptables -L [-t tableName] #Listar todas as regras das cadeas da táboa indicada. Se non se indica táboa por defecto amosa a táboa **filter**.
iptables -L #Listar todas as regras das cadeas da táboa **filter**, é dicir, amosar todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
iptables -L -t nat #Listar todas as regras das cadeas da táboa **nat**, é dicir, amosar todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

- **Táboas/cadeas existentes:**
  - **filter**: Táboa principal para filtrar paquetes. Posúe 3 cadeas:
    - INPUT: Para filtrar paquetes destinados ao noso host local, independentemente da interface ou dirección orixe.
    - FORWARD: Para reenviar paquetes a outro host da rede.
    - OUTPUT: Para filtrar paquetes que saen do host local.
  - **nat**: Soamente para traducir o campo orixe ou destino do paquete (redirección de portos, modificación ip orixe...). Posúe 4 cadeas:
    - PREROUTING: Para alterar paquetes tan pronto como entran.
    - INPUT: Para alterar paquetes destinados ao noso host local.
    - POSTROUTING: Para alterar paquetes cando están a punto de saír do host local.
    - OUTPUT: Para alterar paquetes xerados localmente antes do enrutamento.

      NAT permite traducir direccións de rede da:
      - Rede Interna ➜ Rede Externa(-o) Compartir unha dirección IP Pública (do router) por varios hosts para que poidan ter conexión a internet. Para isto imos empregar a cadea POSTROUTING coa acción (jump=-j) SNAT(se o router posúe IP estática) ou MASQUERADE(se o router posúe IP dinámica). Débese empregar sempre a opción -o (tarxeta de rede de saída).
      - Rede Externa(-i) ➜ Rede Interna(-i) Acceder a un servizo dun host da rede local a través de internet. Para isto imos empregar a cadea PREROUTING coa acción (jump=-j) DNAT(para indicar a IP:Porto do host da rede local que posúe o servizo) ou REDIRECT(para indicar que a redirección ten lugar no propio localhost, é dicir, localhost:80). Así co PREROUTING podemos facer PAT, é dicir, redireccionar portos do router ao host da rede local que nos interese. Débese sempre empregar a opción -i (tarxeta de rede de entrada).

- **mangle**: Para manipular paquetes (TOS, TTL...). Posúe 5 cadeas:
  - PREROUTING: Para manipular os paquetes entrantes antes do enrutamento.
  - INPUT: Para manipular paquetes destinados ao noso host local.
  - FORWARD: Para manipular paquetes que se reenvían a outro host da rede.
  - OUTPUT: Para manipular paquetes xerados localmente antes do enrutamento.
  - POSTROUTING: Para manipular paquetes cando están a punto de saír do host local.

- **raw**: Para marcar os paquetes que non deben ser manexados polo sistema de seguimento de conexións. Posúe 2 cadeas:
  - PREROUTING: Para marcar os paquetes entrantes antes do enrutamento.
  - OUTPUT: Para marcar paquetes xerados localmente antes do enrutamento.

- **security**: Para regras MAC (Mandatory Access Control) (SELinux). Posúe 3 cadeas:
  - INPUT: Regras MAC de paquetes destinados ao noso host local.
  - FORWARD: Regras MAC para reenviar paquetes a outro host da rede.
  - OUTPUT: Regras MAC de paquetes que saen do host local.

NOTAS:

- **Sintaxe (man iptables && man iptables-extensions):**

Se no comando non se especifica a táboa este será executado sempre na táboa filter (táboa considerada predeterminada).

iptables [-t table] {-A|-C|-D|-I} chain rule-specification

rule-specification = [matches...] [target]
match = -m matchname [per-match-options] #match=coincidencia
target = -j targetname [per-target-options] #targetname=destino, como: ACCEPT(aceptar) e DROP(impedir)
-A = engadir como última regra da cadea.
-I = insertar como primeira regra da cadea. Se empregamos -I número, insertarase a regra no número desexado.
-C = comprobar se existe unha regra que coincida coa descrita no comando empregado.
-D = eliminar a regra descrita no comando empregado. Podemos eliminar unha regra dunha cadea indicando o número desta.

iptables [-t table] -P chain target #Cambiar ou determinar a política dunha cadea.
iptables [-t table] {-F|-L|-Z} [chain [rulenum]] [options...]

-F=borrar regras das cadeas.
-L=listar regras da táboa.
-Z=Pór a cero os contadores de paquetes e bytes en todas as cadeas ou na cadea dada.

iptables [-t table] -N chain #Crear unha nova cadea.
iptables [-t table] -X chain #Eliminar cadea.
iptables -L #Listar todas as regras das cadeas da táboa **filter**, é dicir, amosar todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
iptables -L --line-numbers -v#Listar de forma numerada todas as regras das cadeas da táboa **filter** de forma detallada(-v), é dicir, amosar todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT** de forma numerada e detallada(-v), de tal xeito que amosa: nome da interface, opcións da regra se existen, máscaras TOS, contadores de paquetes e bytes.

IMPORTANTE: A opción -v é moi útil para depurar iptables xa que informa sobre contadores de paquetes e bytes na que unha regra vese afectada, é dicir, se unha regra cúmprese no firewall vai recibindo información a cal pode ser revisada coa opción -v.

Resumo Prácticas Exemplos Táboa filter

- No **Exemplo1. Filtrado porto TCP 80 (http)** imos impedir o acceso ao porto TCP 80 (http), é dicir, cortamos o acceso ao servidor Web na máquina virtual A (kaliA).
- No **Exemplo2. Filtrado porto TCP 80 (http) e Prioridade das regras** imos filtrar o acceso ao porto TCP 80 (http), impedindo e aceptando o acceso, según a orde das regras existentes na mesma cadea do firewall iptables.
- No **Exemplo3. Filtrado porto TCP 80 (http) según IP de orixe** imos filtrar o acceso ao porto TCP 80 (http) se a IP de orixe é 192.168.120.101(kaliB), é dicir, imos permitr/cortar o acceso o acceso ao servidor Web na máquina virtual A (kaliA) cando a conexión proveña da IP 192.168.120.101 (kaliB).
- No **Exemplo4. Evitar filtrado porto TCP 80 (http) según IP de orixe** imos modificar a IP da máquina virtual B (kaliB) para poder saltar o firewall iptables.
- No **Exemplo5. Filtrado porto TCP 80 (http) según IP de orixe e MAC-Address de orixe** imos filtrar o acceso ao porto TCP 80 (http) se a IP de orixe e MAC Address corresponden a kaliB, é dicir, imos permitr/cortar o acceso o acceso ao servidor Web na máquina virtual A (kaliA) cando a conexión proveña da IP e MAC Address de kaliB.
- No **Exemplo6. Evitar filtrado porto TCP 80 (http) según IP de orixe e MAC-Address de orixe** imos modificar a IP e MAC Address da máquina virtual B (kaliB) para poder saltar o firewall iptables.
- No **Exemplo7. Filtrado portos TCP 80 (http), 443(https) e 22(ssh)** imos impedir o acceso ao portos TCP 80 (http), 443(https) e 22(ssh), é dicir, cortamos o acceso ao servidor Web e servidor SSH na máquina virtual A (kaliA).
- No **Exemplo8. Filtrado portos TCP 80 (http), 443(https) e 22(ssh) a un rango de IPs orixe** imos impedir o acceso ao portos TCP 80 (http), 443(https) e 22(ssh) da máquina virtual A(kaliA) sempre e cando a conexión veña dun rango de IPs orixe determinado.
- No **Exemplo9. Filtrado portos TCP 80 (http), 443(https) e 22(ssh) a unhas determinadas IPs orixe** imos impedir o acceso ao portos TCP 80 (http), 443(https) e 22(ssh) da máquina virtual A(kaliA) sempre e cando a conexión veña dunhas IPs orixe determinadas.

Resumo Prácticas Exemplos Táboa nat

- No **Exemplo10. Acceso a Internet dende kaliB** imos permitir o enrutamento entre interfaces en kaliA para que kaliB poida ter acceso a Internet, é dicir, imos facer que kaliA simula ser o router facendo NAT para kaliB.
- No **Exemplo11. PAT na intranet** imos redireccionar o porto TCP 8080 ao porto TCP 80 en kaliA se a petición de conectividade realízase dende a rede local 192.168.120.0/24

Firewall - iptables

Máquina virtual A: Kali amd64

1. Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali #Cambiar o contrasinal do usuario kali. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final).

2. Cambiar hostname da máquina virtual A. Por kaliA como hostname:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# echo 'kaliA' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kali:~# echo 'kernel.hostname=kaliA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
   kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.

3. Configurar a rede:
   Na contorna gráfica abrir un terminal e executar:

   kali@kaliA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kaliA:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kaliA:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliA:~# ip addr add 192.168.120.100/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.100 e máscara de subrede: 255.255.255.0.
   root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

4. Comprobar estado do Servidor SSH:
   root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.
   root@kaliA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   root@kaliA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
   root@kaliA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
   root@kaliA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.
   root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.
   root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
   root@kaliA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)
   root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*

root@kaliA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled
root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
root@kaliA:~# ssh -v kali@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario kali e o seu contrasinal. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.
kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.
root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kaliA:~$

Máquina virtual B: Kali amd64

5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kali:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
   root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
   root@kali:~# echo '192.168.120.100 kaliA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kaliA, para que atenda á IP 192.168.120.100
   root@kali:~# ping -c4 kaliA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:
   root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
   kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.

7. **B** $\overset{\text{SSH}}{\rightarrow}$ **A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:
   Na contorna gráfica abrir un terminal e executar:

   kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ nc -vz kaliA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ ssh -v kali@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario **kali** a través da conexión cifrada SSH.
   kali@kaliA:~$

8. Activar Servidor Web Apache:
   kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
   root@kaliA:~# /etc/init.d/apache2 start #Iniciar o servidor web Apache.
   root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
   root@kaliA:~# nc -vz 192.168.120.100 80 #Mediante o comando nc(netcat) comprobar se o porto 80 do servidor web Apache está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto TCP a escanear.

No caso da distribución Kali xa temos instalado o servidor web Apache, pero nunha distribución baseada en Debian poderiamos instalalo do seguinte xeito:

```
# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
# apt search apache2 #Buscar calquera paquete que coincida co patrón de búsqueda apache2
# apt -y install apache2 #Instalar o paquete apache2, é dicir, instalar o servidor HTTP apache2. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

9. Lanzar na máquina virtual B (Kali) un navegador e visitar a IP 192.168.120.100 ou a URL http://192.168.120.100

10. Permisos apache:

```
root@kaliA:~# chown -R www-data. /var/www/html/ #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio DocumentRoot de Apache: /var/www/html
root@kaliA:~# chmod 444 /var/www/html/index.html #Cambiar a só lectura os permisos ugo do ficheiro index.html situado en /var/www/html, é dicir, establecer os permisos r--r--r-- (soamente lectura para o usuario propietario, o grupo propietario e o resto do mundo)
root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.
root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

11. Actualizar na máquina virtual B (Kali) a páxina referente á URL http://192.168.120.100

12. **Exemplo1. Filtrado porto TCP 80 (http)**

Imos engadir na táboa **filter**, na cadea **INPUT** unha regra que denegue **DROP** o acceso ao porto TCP 80 (http)

Procedemento:

1. Engadir a regra:
   root@kaliA:~# iptables -L #Listar todas as regras das cadeas da táboa **filter**, é dicir, amosar todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -j DROP #Denegar acceso ao porto 80 (http)
   root@kaliA:~# iptables -L #Listar todas as regras das cadeas da táboa **filter**, é dicir, amosar todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

2. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora NON seremos quen de visualizar a URL debido á regra iptables xerada.

13. **Exemplo2. Filtrado porto TCP 80 (http) e Prioridade das regras**

Procedemento:

1. Realizar de novo o Exemplo1.

2. Engadir unha nova regra ao final da cadea INPUT:
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT #Aceptar acceso ao porto 80 (http). Coa opción -A a regra engádese como última regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora seguiremos sen poder acceder á URL debido ao funcionamento do firewall. Así, percórrese a cadea INPUT e a primeira regra xa afecta á chamada ao acceso do porto TCP 80, polo que realízase o que a regra determina, neste caso impide o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

4. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -j ACCEPT #Aceptar acceso ao porto 80 (http). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

5. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora SI seremos quen de visualizar a URL debido a que a regra iptables xerada foi insertada como primeira regra da cadea INPUT. Así, percórrese a cadea INPUT e a primeira regra xa afecta á chamada ao acceso do porto TCP 80, polo que realízase o que a regra determina, neste caso permite o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

14. **Exemplo3. Filtrado porto TCP 80 (http) según IP de orixe**

Procedemento:

1. Realizar de novo o Exemplo2.

2. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -s 192.168.120.101 -j DROP #Denegar acceso ao porto 80 (http). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora NON seremos quen de visualizar a URL debido a que a regra iptables xerada foi insertada como primeira regra da cadea INPUT. Así, percórrese a cadea INPUT e a primeira regra cúmprese, polo que realízase o que a regra determina, neste caso denega o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

4. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -s 192.168.120.101 -j ACCEPT #Aceptar acceso ao porto 80 (http). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

5. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora SI seremos quen de visualizar a URL debido a que a regra iptables xerada foi insertada como primeira regra da cadea INPUT. Así, percórrese a cadea INPUT e a primeira regra cúmprese, polo que realízase o que a regra determina, neste caso denega o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

15. **Exemplo4. Evitar filtrado porto TCP 80 (http) según IP de orixe**

Procedemento:

1. Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

2. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -s 192.168.120.101 -j DROP #Denegar acceso ao porto 80 (http). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora NON seremos quen de visualizar a URL debido a que a regra iptables xerada foi insertada como primeira regra da cadea INPUT. Así, percórrese a cadea INPUT e a primeira regra cúmprese, polo que realízase o que a regra determina, neste caso denega o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

4. Modificar a IP de kaliB:
   root@kaliB:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliB:~# ip addr del 192.168.120.101/24 dev eth0 #Eliminar a configuración IP de kaliB, é dicir, eliminar da tarxeta de rede interna eth0 a IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
   root@kaliB:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliB:~# ip addr add 192.168.120.102/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.102 e máscara de subrede: 255.255.255.0.
   root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface eth0 de kaliA

5. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora SI seremos quen de visualizar a URL debido a que ningunha regra iptables cúmprese, e a política por defecto é ACCEPT, co cal a conexión establécese.

16. **Exemplo5. Filtrado porto TCP 80 (http) según IP de orixe e MAC-Address de orixe**

Procedemento:

1. Realizar de novo o Exemplo4.

2. Descubrir a MAC-Address da interface eth0 (192.168.100.102) de KaliB.
   root@kaliA:~# ping -c4 192.168.120.102 #Comprobar mediante o comando ping a conectividade coa interface eth0 de kaliB
   root@kaliA:~# arp #Revisar a táboa arp, é dicir, visualizar a caché arp do sistema: asignación IP coa súa correspondente dirección física (MAC Address). Podemos observar si existe unha entrada para kaliB, onde se asigna a IP 192.168.120.102 a súa dirección física (MAC Address ou HWaddress).
   root@kaliA:~# MAC_kaliB=$(arp -n | grep 192.168.120.102 | awk '{print $3}')#Gardar a MAC-Address atopada na variable MAC_kaliB

   > # dpkg -l | grep net-tools ; [$(echo $?) -eq '1'] && apt update && apt -y install net-tools #Verificar se o paquete net-tools está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase. O paquete net-tools é necesario para poder empregar comandos coma: ifconfig, netstat, route e arp.

3. Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

4. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -I INPUT -p tcp --dport 80 -s 192.168.120.102 -m mac --mac-source ${MAC_kaliB} -j DROP #Denegar acceso ao porto 80 (http). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

5. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   > Agora NON seremos quen de visualizar a URL debido a que a regra iptables xerada foi insertada como primeira regra da cadea INPUT. Así, percórrese a cadea INPUT e a primeira regra cúmprese, polo que realízase o que a regra determina, neste caso denega o acceso. Deste xeito, como xa unha regra no firewall executouse remátase o filtrado e non se lee ningunha regra mais.

17. **Exemplo6. Evitar filtrado porto TCP 80 (http) según IP de orixe e MAC-Address de orixe**

Procedemento:

1. Realizar de novo o Exemplo5.

2. Modificar a IP de kaliB:

   root@kaliB:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliB:~# ip addr del 192.168.120.102/24 dev eth0 #Eliminar a configuración IP de kaliB, é dicir, eliminar da tarxeta de rede interna eth0 a IP: 192.168.120.102 e máscara de subrede: 255.255.255.0.
   root@kaliB:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kaliB:~# ip addr add 192.168.120.103/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.103 e máscara de subrede: 255.255.255.0.
   root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

3. Modificar a MAC-Address de kaliB:

   IMPORTANTE:
   Como máquina virtual teremos un problema cando modifiquemos a mac-address, tal que perderemos a comunicación de rede, posto que as máquinas virtuais están configuradas cunha mac-address e non permiten a comunicación se cambiamos esta por software. Habería que apagar a máquina virtual e modificar a mac-address polo que queiramos.

   Opción 1

   root@kaliB:~# macchanger -l #Listar posibles 3 primeiros octetos de MAC Address de coñecidos proveedores.
   root@kaliB:~# ip link set dev eth0 down #Deshabilitar interface eth0.
   root@kaliB:~# macchanger -m 08:00:46:44:55:66 eth0 #Cambiar a MAC-Address da interface eth0 a 08:00:46:44:55:66.
   root@kaliB:~# ip link set dev eth0 up #Habilitar interface eth0.
   root@kaliB:~# ip addr show eth0 #Amosar a configuración da tarxeta de rede eth0.
   root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

   Opción 2

   root@kaliB:~# ip link set dev eth0 down #Deshabilitar interface eth0.
   root@kaliB:~# ip link set dev eth0 address 08:00:46:44:55:66 eth0 #Cambiar a MAC-Address da interface eth0 a 08:00:46:44:55:66.
   root@kaliB:~# ip link set dev eth0 up #Habilitar interface eth0.
   root@kaliB:~# ip addr show eth0 #Amosar a configuración da tarxeta de rede eth0.
   root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

4. Acceder de novo dende o equipo cliente kaliB á URL http://192.168.120.100

   Agora SI seremos quen de visualizar a URL debido a que ningunha regra iptables cúmprese, e a política por defecto é ACCEPT, co cal a conexión establécese.

18. **Exemplo7. Filtrado portos TCP 80 (http), 443(https) e 22(ssh)**

Imos insertar na táboa **filter**, na cadea **INPUT** unha regra que denegue **DROP** o acceso aos portos TCP 80 (http), 443(https) e 22(ssh).

Procedemento:

1. Realizar de novo o Exemplo6.

2. Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -I INPUT -p tcp -m multiport --dports 80,443,22 -s 192.168.120.103 -j DROP #Denegar acceso aos portos 80 (http), 443 (https) e 22 (ssh). Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

4. Acceder de novo dende o equipo cliente kaliB:
   - Á URL http://192.168.120.100
   - Á URL https://192.168.120.100
   - Ao servidor ssh 192.168.120.100
     kali@kaliB:~$ ssh kali@192.168.120.100

     Agora NON seremos quen de visualizar as URLs nin acceder ao servidor SSH debido á regra iptables xerada.

19. **Exemplo8. Filtrado portos TCP 80 (http), 443(https) e 22(ssh) a un rango de IPs orixe**

Imos insertar na táboa **filter**, na cadea **INPUT** unha regra que denegue **DROP** o acceso aos portos TCP 80 (http), 443(https) e 22(ssh) ás IPs orixe 192.168.120.101, 192.168.120.102 e 192.168.120.103.

Procedemento:

1. Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

2. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -I INPUT -p tcp -m multiport --dports 80,443,22 -m iprange --src-range 192.168.120.101-192.168.120.103 -j DROP #Denegar acceso aos portos 80 (http), 443 (https) e 22 (ssh) para o rango de IPs dende 192.168.120.101 ata 192.168.120.103. Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Acceder de novo dende o equipo cliente kaliB:
   - Á URL http://192.168.120.100
   - Á URL https://192.168.120.100
   - Ao servidor ssh 192.168.120.100
     kali@kaliB:~$ ssh kali@192.168.120.100

     Agora NON seremos quen de visualizar as URLs nin acceder ao servidor SSH debido á regra iptables xerada.

20. **Exemplo9. Filtrado portos TCP 80 (http), 443(https) e 22(ssh) a unhas determinadas IPs orixe**

Imos insertar na táboa **filter**, na cadea **INPUT** unha regra que denegue **DROP** o acceso aos portos TCP 80 (http), 443(https) e 22(ssh) ás IPs orixe 192.168.120.101, 192.168.120.102 e 192.168.120.103.

Procedemento:

1. Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

2. Insertar unha nova regra no inicio da cadea INPUT:
   root@kaliA:~# iptables -I INPUT -p tcp -m multiport --dports 80,443,22 -s 192.168.120.101,192.168.120.102,192.168.120.103 -j DROP #Denegar acceso aos portos 80 (http), 443 (https) e 22 (ssh) para as IPs 192.168.120.101, 192.168.120.102 e 192.168.120.103. Coa opción -I a regra insértase como a primeira regra da cadea correspondente.
   root@kaliA:~# iptables -L --line-numbers #Listar de forma numerada todas as regras das cadeas da táboa **filter**, é dicir, amosar de forma numerada todas as regras das cadeas **INPUT**, **FORWARD** e **OUTPUT**.

3. Acceder de novo dende o equipo cliente kaliB:
   - Á URL http://192.168.120.100
   - Á URL https://192.168.120.100
   - Ao servidor ssh 192.168.120.100
     kali@kaliB:~$ ssh kali@192.168.120.100

     Agora NON seremos quen de visualizar as URLs nin acceder ao servidor SSH debido á regra iptables xerada.

21. **Exemplo10. Acceso a Internet dende kaliB**

Imos permitir o enrutamento entre interfaces en kaliA para que kaliB poida ter acceso a Internet, é dicir, imos facer que kaliA simula ser o router facendo NAT para kaliB.

Procedemento:

1. Permitir o enrutamento entre interfaces en kaliA.

   Opción 1: De forma temporal modificando /proc/sys/net/ipv4/ip_forward

   root@kaliA:~# echo 1 > /proc/sys/net/ipv4/ip_forward #Activar enrutamento entre interfaces, é dicir, permitir que pasen paquetes entre eth0(rede interna) e eth1(saída a internet)

   Opción 2: De forma permanente modificando /etc/sysctl.conf

   root@kaliA:~# echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf #Activar enrutamento entre interfaces, é dicir, permitir que pasen paquetes entre eth0(rede interna) e eth1(saída a internet)
   root@kaliA:~# sysctl -p #Activar o cambio realizado en /etc/sysctl.conf sen ter que pechar sesión nin reiniciar

2. Eliminar todas as regras de todas as cadeas das táboas filter e nat.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -F -t nat #Eliminar todas as regras de todas as cadeas da táboa nat.
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

3. Insertar unha nova regra no inicio da cadea POSTROUTING:

   Opción 1: Co salto SNAT xa que eth1 en VirtualBox imos considerar que posúe a IP estática 10.0.3.15

   root@kaliA:~# iptables -t nat -I POSTROUTING -s 192.168.120.101,192.168.120.102,192.168.120.103 -o eth1 -j SNAT --to 10.0.3.15 #Permitir acceso a Internet facendo NAT aos hosts que posúan calquera das IPs 192.168.120.101, 192.168.120.102, 192.168.120.103. Para permitir o acceso créase esta regra nat que enruta todos os paquetes desas IPs á tarxeta eth1 que posúe a IP estática 10.0.3.15.
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

   Opción 2: Co salto MASQUERADE xa que eth1 en VirtualBox imos considerar que posúe a IP dinámica 10.0.3.15

   root@kaliA:~# iptables -t nat -I POSTROUTING -s 192.168.120.101,192.168.120.102,192.168.120.103 -o eth1 -j MASQUERADE #Permitir acceso a Internet facendo NAT aos hosts que posúan calquera das IPs 192.168.120.101, 192.168.120.102, 192.168.120.103. Para permitir o acceso créase esta regra nat que enruta todos os paquetes desas IPs á tarxeta eth1 que posúe a IP dinámica 10.0.3.15.
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

4. Configurar a rede en kaliB: táboa de enrutamento e configuración DNS
   root@kaliA:~# ip route add default via 192.168.120.100 dev eth0 #Pór a IP de kaliA como gateway por defecto na táboa de rutas de kaliB
   root@kaliA:~# echo 'nameserver 8.8.8.8' >> /etc/resolv.conf #Pór como servidor DNS primario 8.8.8.8 en kaliB

5. Probar o acceso a Internet dende kaliB accedendo á URL www.debian.org:

   kaliB accede a internet xa que temos unha regra nat en kaliA que lle permite o acceso.

22. **Exemplo11. PAT na intranet**

Imos redireccionar o porto TCP 8080 ao porto TCP 80 en kaliA se a petición de conectividade realízase dende a rede local 192.168.120.0/24

Procedemento:

1. Permitir o enrutamento entre interfaces en kaliA.

   Opción 1: De forma temporal modificando /proc/sys/net/ipv4/ip_forward

   root@kaliA:~# echo 1 > /proc/sys/net/ipv4/ip_forward #Activar enrutamento entre interfaces, é dicir, permitir que pasen paquetes entre eth0(rede interna) e eth1(saída a internet)

   Opción 2: De forma permanente modificando /etc/sysctl.conf

   root@kaliA:~# echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf #Activar enrutamento entre interfaces, é dicir, permitir que pasen paquetes entre eth0(rede interna) e eth1(saída a internet)
   root@kaliA:~# sysctl -p #Activar o cambio realizado en /etc/sysctl.conf sen ter que pechar sesión nin reiniciar

2. Eliminar todas as regras de todas as cadeas das táboas filter e nat.
   root@kaliA:~# iptables -F #Eliminar todas as regras de todas as cadeas da táboa filter.
   root@kaliA:~# iptables -F -t nat #Eliminar todas as regras de todas as cadeas da táboa nat.
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

3. Insertar unha nova regra no inicio da cadea PREROUTING:

   Opción 1: Co salto DNAT na interface eth0 para ter soamente en conta as peticións da rede 192.168.120.0/24

   root@kaliA:~# iptables -t nat -I PREROUTING -s 192.168.120.0/24 -p tcp --dport 8080 -i eth0 -j DNAT --to 192.168.120.100:80 #Redireccionar en kaliA calquera chamada dende a rede 192.168.120.0/24 o porto TCP 8080 ao porto TCP 80
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

   Opción 2: Co salto REDIRECT xa que o PAT faise dentro do mesmo host(kaliA) na interface eth0 para ter soamente en conta as peticións da rede 192.168.120.0/24

   root@kaliA:~# iptables -t nat -I PREROUTING -s 192.168.120.0/24 -p tcp --dport 8080 -i eth0 -j REDIRECT --to 80 #Redireccionar en kaliA calquera chamada dende a rede 192.168.120.0/24 o porto TCP 8080 ao porto TCP 80
   root@kaliA:~# iptables -L --line-numbers -t nat#Listar de forma numerada todas as regras das cadeas da táboa **nat**, é dicir, amosar de forma numerada todas as regras das cadeas **PREROUTING**, **INPUT**, **POSTROUTING** e **OUTPUT**.

4. Configurar a rede en kaliB: táboa de enrutamento e configuración DNS
   root@kaliA:~# ip route add default via 192.168.120.100 dev eth0 #Pór a IP de kaliA como gateway por defecto na táboa de rutas de kaliB
   root@kaliA:~# echo 'nameserver 8.8.8.8' >> /etc/resolv.conf #Pór como servidor DNS primario 8.8.8.8 en kaliB

5. Probar o acceso dende o equipo cliente kaliB á URL http://192.168.120.100:8080

   kaliB accede á paxina web http://192.168.120.100:80 xa que realizouse a redirección do porto TCP 8080 ao porto TCP 80.

Ricardo Feijoo Costa