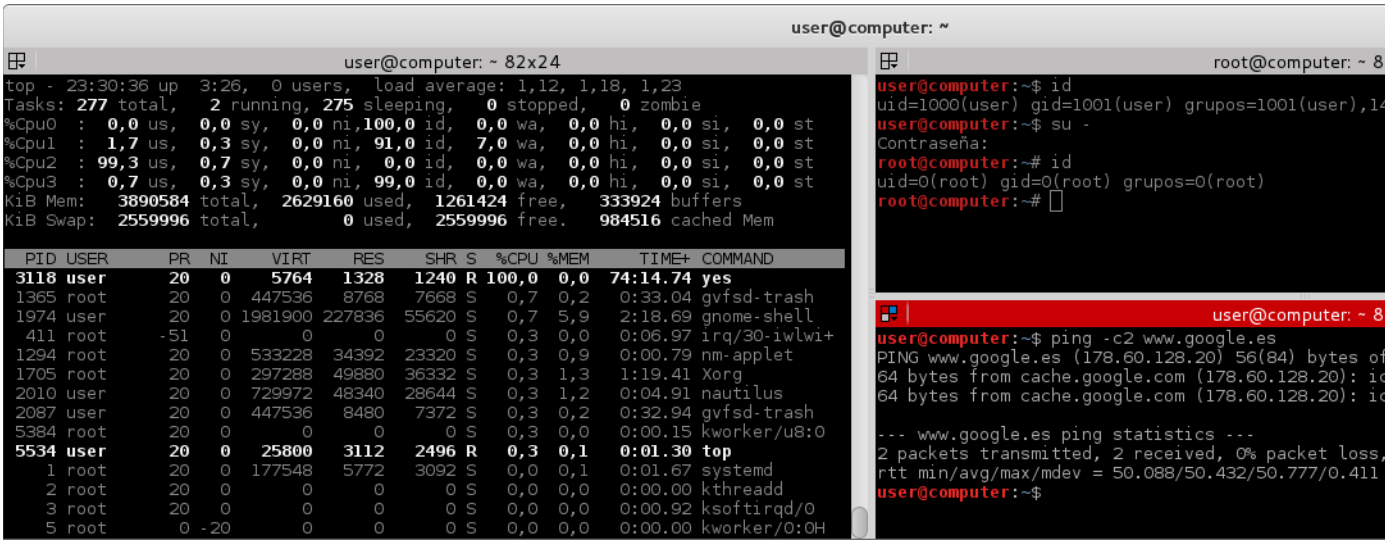
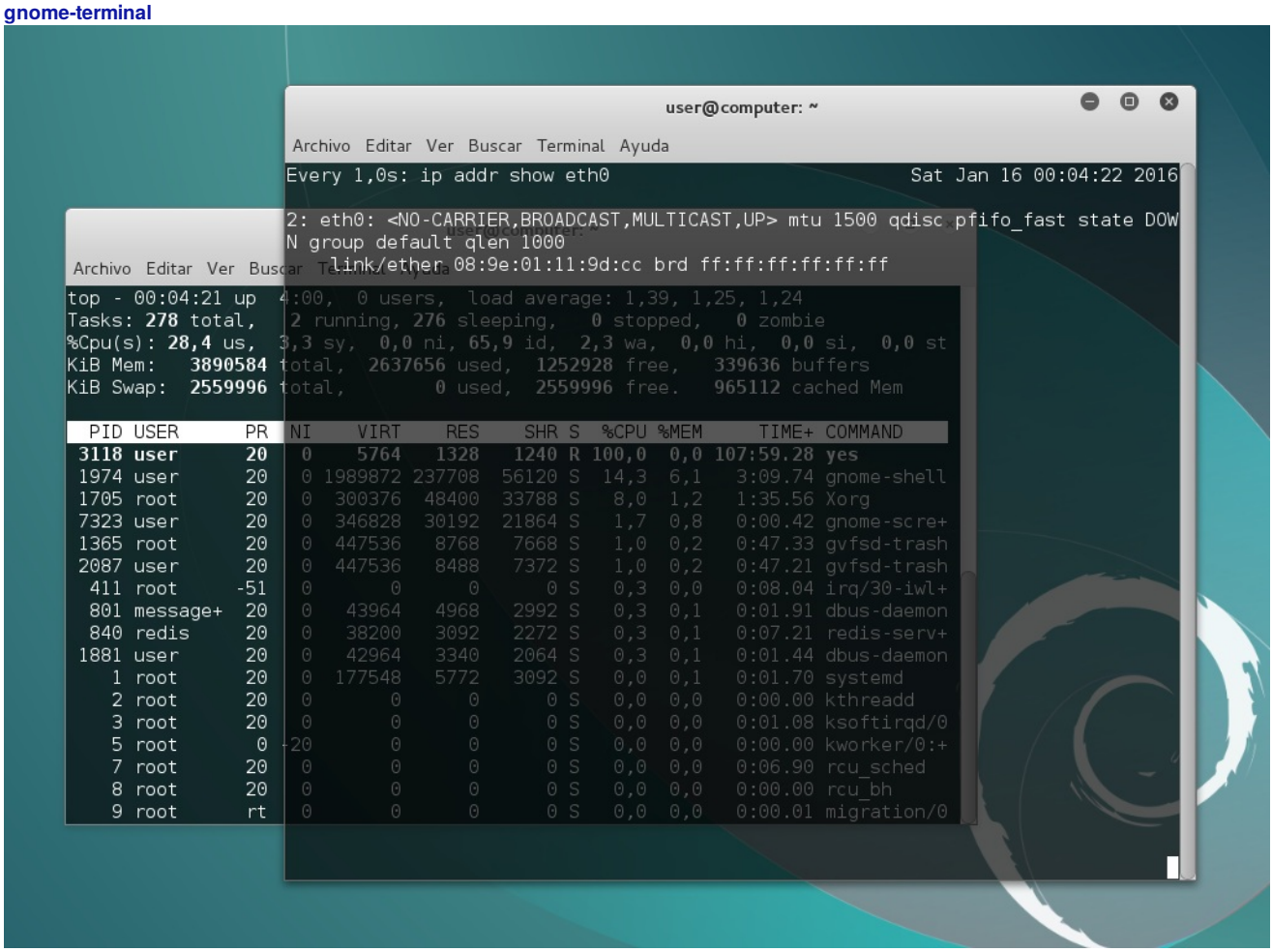
Comandos GNU/Linux e SHELL BASH (/bin/bash)

terminator



gnome-terminal



Ricardo Feijoo Costa

Servizos GNU/Linux

- **Instalar servizo (como outro paquete calquera)**:
  # apt update || apt-get update #Actualizar repositorios declarados no ficheiro */etc/apt/souces.list* e nos ficheiros existentes no directorio */etc/apt/sources.list.d*
  Así, unha vez realizada a consulta dos ficheiros existentes nas rutas anteriores, descárganse uns ficheiros coas listas de paquetes posibles a instalar. Estes ficheiros son gardados en */var/lib/apt/lists*
  # apt search pattern || apt-cache search pattern #Buscar nas anteriores listas descargadas en */var/lib/apt/lists* paquetes que coincidan co patrón de búsqueda *pattern*. A saída do/s comando/s amosan o nome do/s paquete/s e unha pequena descrición do/s mesmo/s.

    O comando non precisa ser lanzado con pemisos de *root* xa que o directorio */var/lib/apt/lists* ten acceso de lectura e execución para calquera usuario do sisetema, e os ficheiros que aí se atopan tamén teñen permisos de lectura para calquera usuario do sistema. Polo tanto, unha vez realizada a actualización dos repositorios, un usuario sen permisos de *root* podería executar:
    $ apt search pattern || apt-cache search pattern

  # apt -y install packageName || apt-get -y install packageName #Instalar un Servizo mediante a instalación do paquete de nome *packageName*. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

- **Accións(action) sobre o servizo**: As accións que normalmente se poden realizar cun servizo son: {start|stop|reload|force-reload|restart|try-restart|status}
  - **start** ➜ arrancar
  - **stop** ➜ parar
  - **reload** ➜ recargar o ficheiro de configuración
  - **force-reload** ➜ recargar se o servizo o permite o ficheiro de configuración. Se non o permite reinicia o servizo para que teña lugar a nova configuración
  - **restart** ➜ reiniciar, é dicir, paro o servizo no caso que esté arrancado, e arráncao. Se non está arrancado soamente o arranca.
  - **try-restart** ➜ reiniciar, é dicir, reinicia o servizo se este xa está arrancado. No caso que non estea arrancado non fai nada.
  - **status** ➜ revisar o estado

  Estas accións poder executarse con diversa sintaxe según o comando empregado. Así, considerando o nome do servizo como *serviceName* e a acción a realizar como *action*, poderían executarse as accións como segue:

  - Mediante a execución **/etc/init.d**:
    # /etc/init.d/serviceName action
  - Mediante a execución **service**:
    # service serviceName action
  - Mediante a execución **invoke-rc.d**:
    # invoke-rc.d serviceName action
  - Mediante a execución **systemctl**:
    # systemctl action serviceName

- **Desinstalar servizo (como outro paquete calquera)**:
  # apt -y remove packageName || apt-get -y remove packageName #Eliminar un Servizo mediante a desinstalación do paquete de nome *packageName*. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na desinstalación do paquete. **IMPORTANTE:** Con **remove NON SE ELIMINAN os ficheiros de configuración** do paquete desinstalado.
  # apt -y purge packageName || apt-get -y purge packageName #Eliminar un Servizo mediante a purga do paquete de nome *packageName*. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na desinstalación do paquete. **IMPORTANTE:** Con **purge SI SE ELIMINAN os ficheiros de configuración** do paquete desinstalado.

Ricardo Feijoo Costa

Sintaxe Ficheiros configuración servizos GNU/Linux

NOTAS:

Recén instalado un servizo o ficheiro de configuración deste servizo soe posuír a seguinte sintaxe:

- As liñas que comezan co caracter **#** e as liñas en branco son consideradas comentarios.
- Aparecen por liña pares de valores: directiva e argumento.
- As directivas a configurar aparecen comentadas.
- As directivas configuradas (activadas) aparecen descomentadas.
- Cando unha directiva non aparece descomentada e si comentada soe indicar que está activa e determina a configuración por defecto do servizo.

Para activar unha directiva hai que modificala, engadila ou descomentala e recargar o servizo en cuestión:

 # /etc/init.d/servizo reload

Recargar o ficheiro de configuración mediante a sintaxe **/etc/init.d/servizo reload**:

1. Permite recargar a configuración do servizo **sen interromper** as conexións establecidas co mesmo.
2. Pero, os cambios afectarán ás novas conexións e non as que están activas.

Tamén se pode recargar o ficheiro de configuración mediante a sintaxe **/etc/init.d/servizo restart**, pero a execución deste comando:

1. Permite recargar a configuración do servizo **interrompendo** as conexións establecidas co mesmo.
2. Os cambios debido á interrupción afectarán a todas as conexións.

Ricardo Feijoo Costa

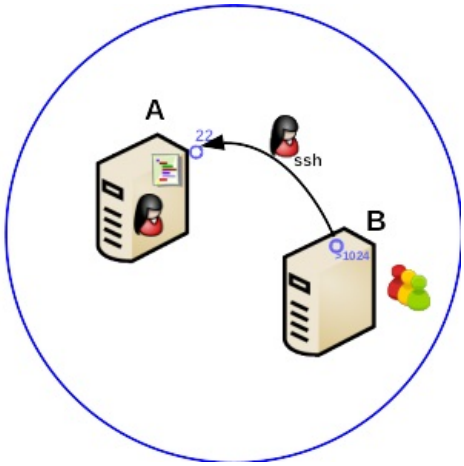
Conexión Remota mediante SSH
Cambios en ssh_config

ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede Interna
Servidor SSH: openssh-server
ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
ISO: Kali Live amd64
IP/MS: 192.168.120.101/24



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:

- **Cliente ssh GNU/Linux:**
  - Comando ssh. Paquete openssh-client (# apt update && apt -y install openssh-client).
  - Configuración **(man ssh_config)**:

    O cliente (comando ssh) posúe unha configuración predeterminada que podemos modificar. A orde de prioridade desa configuración é:

    1. Opcións invocadas dende a liña de comandos ao executar o propio comando ssh co parámetro **-o**
    2. Opcións invocadas a través do ficheiro pertencente a cada usuario situado na ruta **~/.ssh/config**
    3. Opcións invocadas a través do ficheiro de configuración global do sistema en **/etc/ssh/ssh_config**

    Nos ficheiros de configuración (~/.ssh/config e /etc/ssh/ssh_config):

    - Aparecen por liña pares de valores: directiva e argumento. As directivas non distinguen entre maiúsculas e minúsculas e os argumentos si.
    - As liñas que comezan co caracter **#** e as liñas en branco son consideradas comentarios.
    - É soamente cambiada a primeira vez que aparece. Así, definicións específicas de host deberían estar no comezo da configuración dos ficheiros e as opcións por defecto ao final. Dese xeito no caso de repetir unha configuración terase en conta a primeira, que seguramente sexa o que queiramos.

    Host ➜ Restrinxe as seguintes directivas existentes ata atopar outra directiva Host. Admite máis de 1 argumento separados polo caracter espazo. Como argumento admite:

    StrictHostKeyChecking ➜ Directiva que determina se se confía na key host do servidor SSH co que se establece a conexión. Os hosts keys son gardados por defecto no ficheiro ~/.ssh/known_hosts.

    User ➜ Directiva que determina o usuario que fai login para establecer a conexión.

    Port ➜ Directiva que determina o porto TCP co que establecer conexión no servidor SSH.

NOTAS:

- **Formato comandos conexión SSH:**
  - Comando ssh: Consola remota ou execución remota de comandos.

    $ ssh [-p port] user@hostname [command] || ssh [-p port] -l user hostname [command]

    ssh ➜ comando cliente para realizar as conexións SSH. Execútase no equipo cliente.

    -p port ➜ indica o porto TCP(ver /etc/services) onde espera as conexións o servidor SSH. É opcional e se non se especifica indica que a conexión terá lugar por defecto no porto TCP 22(ver /etc/services), a non ser que apareza configurado a directiva Port no arquivo *~/.ssh/config* ou no arquivo */etc/ssh/ssh_config*

    user@hostname ➜ indica o usuario (user) co que se quere establecer a conexión e o hostname onde espera o servidor ssh. Pode empregarse outra nomenclatura equivalente: *-l user hostname* para facer login co usuario user.

    user ➜ indica o usuario co que se quere acceder ao servidor SSH, o cal debe existir no servidor SSH.

    hostname ➜ indica o nome do servidor SSH. Pode tomar o valor:

    - Do nome configurado no arquivo /etc/hosts do equipo cliente
    - Da IP
    - Do nome DNS

    command ➜ indica o/s comando/s (script) a executar no servidor SSH, amosando a saída do/s comando/s na máquina cliente. É opcional, polo que se non se especifica abrirase unha consola de conexión remota co servidor SSH.

  - Comando scp: Copia remota segura.

    $ scp [-P port] user@hostname:remote_path local_path #Copiar ficheiros
    $ scp -r [-P port] user@hostname:remote_path local_path #Copiar directorios recursivamente
    $ scp [-P port] local_path user@hostname:remote_path #Copiar ficheiros
    $ scp -r [-P port] local_path user@hostname:remote_path #Copiar directorios recursivamente

    scp ➜ comando cliente para realizar as copias seguras mediante conexións SSH. Execútase no equipo cliente.

    -P port ➜ indica o porto TCP(ver /etc/services) onde espera as conexións o servidor SSH. É opcional e se non se especifica indica que a conexión terá lugar por defecto no porto TCP 22(ver /etc/services), a non ser que apareza configurado a directiva Port no arquivo *~/.ssh/config* ou no arquivo */etc/ssh/ssh_config*

    user@hostname ➜ indica o usuario (user) co que se quere establecer a conexión e o hostname onde espera o servidor ssh.

    user ➜ indica o usuario co que se quere acceder ao servidor SSH, o cal debe existir no servidor SSH.

    hostname ➜ indica o nome do servidor SSH. Pode tomar o valor:

    - Do nome configurado no arquivo /etc/hosts do equipo cliente
    - Da IP
    - Do nome DNS

    local_path ➜ indica a ruta local(ruta do cliente) a copiar ou onde volcar o copiado

    :remote_path ➜ indica a ruta remota(ruta do servidor) onde copiar ou de onde copiar. Se non se especifica remote_path por defecto ou especifícase unha ruta relativa o caracter ':' simboliza a variable $HOME do usuario co que se establece a conexión. En caso contrario pode especificarse unha ruta absoluta deixando de ter valor o caracter ':'

    -r ➜ permite copiar directorios enteiros recursivamente. A opción *-r* segue ligazóns simbólicas.

NOTAS:

- **Servidor ssh GNU/Linux**:
  - Paquete openssh-server (# apt update && apt -y install openssh-server).
  - Ficheiro de configuración: **/etc/ssh/sshd_config (man sshd_config)**

O **servizo SSH** permite obter, mediante conexión cifrada, un terminal de comandos a quen accede de forma remota. Os comandos tamén poden ser aplicacións gráficas xa que o **servizo SSH** tamén permite redireccionar o servidor gráfico, de tal xeito que un comando que emprega librarías gráficas, como por exemplo Firefox, será visionado na máquina cliente (a que accede ao servizo SSH).

O **porto TCP**(ver /etc/services) de conexión por defecto é o **22**, pero pódese configurar.

Para conectarse é necesario posuir un **cliente ssh**, tipicamente o comando **ssh**, o cal soe vir preinstalado por defecto nas distribucións GNU/Linux.

O arquivo de configuración de sistema do servidor ssh pódese atopar na ruta:
/etc/ssh/sshd_config

Práctica - Conexión Remota mediante SSH - Cambios en ssh_config

Máquina virtual A: Kali amd64

1. Pór contrasinal ao usuario, co que arranca a live, de nome: kali
   Na contorna gráfica abrir un terminal e executar:

   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali || (echo -e 'kali\nabc123.\nabc123.' | passwd) #Cambiar o contrasinal do usuario kali. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final).
   kali@kali:~$ sudo passwd root || (sudo -c "echo -e 'abc123.\nabc123.' | passwd") #Cambiar o contrasinal do usuario root. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final). O cambio de contrasinal é posible debido aos permisos configurados co comando sudo (/etc/sudoers, visudo).

2. Cambiar hostname da máquina virtual A. Por kaliA como hostname:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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:~$ 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:~#

Máquina virtual B: Kali amd64

5. Pór contrasinal ao usuario, co que arranca a live, de nome: kali

   Na contorna gráfica abrir un terminal e executar:

   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali || (echo -e 'kali\nkaliBpass\nkaliBpass' | passwd) #Cambiar o contrasinal do usuario kali. Por como contrasinal **kaliBpass**

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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. Configurar a rede:

   Na contorna gráfica abrir un terminal e executar:

   kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kaliB:~# /etc/init.d/avahi-daemon stop || systemctl stop avahi-daemon #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@kaliB:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@kaliB:~# 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@kaliB:~# 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@kaliB:~# 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@kaliB:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
   root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa Máquina Virtual A na IP 192.168.120.100
   root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

8. Comprobar estado do Servidor SSH

   kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh (192.168.120.100) 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 kali@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el en 192.168.120.100 co usuario kali e o seu contrasinal no porto TCP 22. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.
   kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **kali** en **kaliB**.
   kali@kaliB:~$

9. Execución de comandos a través da conexión SSH:

- HostA: Servidor SSH coa IP 192.168.120.100
- HostB: Cliente SSH coa IP 192.168.120.101

SSH
B ➜ A Acceder mediante SSH dende o HostB (cliente SSH) ao HostA (servidor SSH):

kali@kaliB:~$ ssh kali@192.168.120.100 #Acceder de B a A mediante o usuario de nome kali ao porto TCP 22(ver /etc/services) onde se supón que está esperando o servizo SSH do HostA, a non ser que teñamos configurado no equipo cliente o arquivo *~/.ssh/config* ou o arquivo */etc/ssh/ssh_config* coa directiva *Port* indicando un porto distinto.
kali@kaliB:~$ ssh -l kali 192.168.120.100 #Comando equivalente ao anterior.
kali@kaliB:~$ ssh kali@192.168.120.100 -p 22 #Acceder de B a A mediante o usuario de nome kali ao porto TCP 22 do servidor SSH.
kali@kaliB:~$ ssh -p 22 -l kali 192.168.120.100 #Comando equivalente ao anterior.
kali@kaliB:~$ ssh 192.168.120.100 #Acceder de B a A mediante o usuario co que estamos conectados no sistema do HostB (neste caso como indica o prompt PS1 intentaríase a conexión co usuario kali), a non ser que tiñamos configurado no arquivo *~/.ssh/config* ou no arquivo */etc/ssh/ssh_config* do equipo cliente a directiva *User* cun nome de usuario. Como non se indica o porto TCP da conexión, esta terá lugar no porto TCP 22(ver /etc/services), a non ser que teñamos configurado no equipo cliente o arquivo *~/.ssh/config* ou o arquivo */etc/ssh/ssh_config* coa directiva *Port* indicando un porto distinto.
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/motd #Executar o comando *cat /etc/motd* no servidor SSH 192.168.120.100(hostA) e ver a saída da execución do comando na máquina local kaliB(hostB).
kali@kaliB:~$ ssh kali@192.168.120.100 "netstat -natp | grep 22" #Executar o comando *netstat* no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB.
kali@kaliB:~$ ssh kali@192.168.120.100 ss -natp #Executar o comando *ss* no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB.
kali@kaliB:~$ scp kali@192.168.120.100:/etc/passwd . #Estando situado no HostB, copiar de A a B (do servidor ao cliente) o arquivo /etc/passwd, é dicir, copiar en B o ficheiro /etc/passwd existente no HostA, e copialo na ruta onde lanza o comando o usuario cliente que é o que simboliza o caracter '.' . Neste caso a copia realizarase no $HOME(~) do usuario local /home/kali
kali@kaliB:~$ echo $(hostname) > fileHostB.txt && scp ./fileHostB.txt kali@192.168.120.100: #Crear en B o ficheiro /home/kali/filehostB.txt co contido de 1 liña que posúe o nome do equipo: kaliB. Se este comando non obtivo erro na súa execución (&&) execútase o segundo comando, o cal indica que estando situado no HostB, copiar de B a A o arquivo /home/kali/fileHostB.txt na casa do usuario kali do hostA, é dicir, copiar en A o ficheiro filehostB.txt no cartafol /home/kali, que é o que simboliza o caracter ':'
kali@kaliB:~$ scp -r kali@192.168.120.100:/tmp . #Estando situado no HostB, copiar de A a B (do servidor ao cliente) todo o directorio /tmp, é dicir, copiar en B o directorio /tmp (e todo que colga deste) existente no HostA, e copialo na ruta onde lanza o comando o usuario cliente que é o que simboliza o caracter '.' . Neste caso a copia realizarase no $HOME(~) do usuario local /home/kali
kali@kaliB:~$ mkdir ~/cousas && cp -pv /etc/passwd cousas #No hostB(cliente) crear o directorio /home/kali/cousas. Se este comando non obtivo erro na súa execución (&&) execútase o segundo comando, o cal indica copiar en modo verbose (detallado) e preservando permisos e datas o ficheiro /etc/passwd dentro de /home/kali/cousas.
kali@kaliB:~$ scp -r cousas kali@192.168.120.100:/tmp #Estando situado no HostB, copiar de B a A (do cliente ao servidor) todo o directorio cousas dentro do directorio /tmp do servidor(hostA), é dicir, copiar en A o directorio /home/kali/cousas (e todo que colga deste) existente no HostB, e copialo dentro da ruta /tmp do servidor
kali@kaliB:~$ ln -s /tmp cousas/tmp #No hostB(cliente) crear o enlace simbólico /home/kali/cousas/tmp que apunta a /tmp
kali@kaliB:~$ scp -r cousas kali@192.168.120.100:/tmp #Estando situado no HostB, copiar de B a A (do cliente ao servidor) todo o directorio cousas dentro do directorio /tmp do servidor(hostA), é dicir, copiar en A o directorio /home/kali/cousas (e todo que colga deste) existente no HostB, e copialo dentro da ruta /tmp do servidor. Como agora dentro de cousas existe un enlace simbólico a /tmp, non se copia o enlace simbólico, senón que se segue ese enlace e polo tanto copiase tamén no servidor(hostA) o contido do directorio /tmp do cliente(hostB) en /home/kali/cousas/tmp, é dicir:

- No cliente(hostB) cousas/tmp é unha ligazón simbólica a /tmp
- No servidor(hostA) cousas/tmp é un directorio coa copia do contido do cartafol /tmp do cliente(hostB)

10. ssh_config: Modificar a configuración de acceso ao servidor SSH de determinados hosts.

Host ➜ Restrinxe as seguintes directivas existentes ata atopar outra directiva Host ou Match. Admite máis de 1 argumento separados polo caracter espazo. Como argumento admite:

- ***** ➜ Simboliza todos os hosts
- **?** ➜ Simboliza un caracter
- **hostname** ➜ Un host determinado
- **IP** ➜ Un host determinado
- ***.example.local** ➜ Calquera host do dominio example.local
- **192.168.120.10?** ➜ Calquera host que coincida no rango 192.168.120.10[0-9]
- **!argumento** ➜ Nega o argumento, indicando que as directivas desta sección Host non terán lugar nese argumento.

StrictHostKeyChecking ➜ Directiva que determina se se confía na key host do servidor SSH co que se establece a conexión. Os hosts keys son gardados por defecto no ficheiro ~/.ssh/known_hosts. Pode tomar como argumento:

- **no** ➜ Garda automaticamente a host key do servidor SSH no ficheiro ~/.ssh/known_hosts
- **off** ➜ Garda automaticamente a host key do servidor SSH no ficheiro ~/.ssh/known_hosts
- **ask** ➜ Pregunta se se confía na host key do servidor SSH co que se establece a conexións. Se respostamos *yes* gardarase a host key no ficheiro *~/.ssh/known_hosts* e a conexión terá lugar. No caso de respostar *non* a conexión non se establecerá.
- **yes** ➜ Nunca engade automaticamente a host key do servidor SSH no ficheiro ~/ssh/known_hosts, e rechaza conectar a hosts que cambiaran a host key.
- **accept-new** ➜ Garda automaticamente a host key do servidor SSH no ficheiro ~/.ssh/known_hosts, e rechaza conectar a hosts que cambiaran a host key.

kali@kaliB:~$ rm -f ~/.ssh/known_hosts #Eliminar sen pedir confirmación o ficheiro ~/.ssh/known_hosts, no cal gárdanse as hosts keys dos servidores coñecidos tras conexións SSH.
kali@kaliB:~$ cat > ~/.ssh/config <<EOF #Comezo do ficheiro a crear ~/.ssh/config
Host * #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
StrictHostKeyChecking no #Gardar automaticamente en ~/.ssh/known_hosts a host key do servidor a conectar.
EOF Fin do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ ssh -o StrictHostKeyChecking=ask kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión no porto TCP por defecto 22 e co usuario kali. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro *~/.ssh/config*, pero neste caso introdúcese por liña de comandos mediante a opción *-o* que o *StrictHostKeyChecking* está a *ask*. Entón pregúntase pola confirmación do host key do servidor gardándose para establecer a conexión e poder gardala no ficheiro *~/.ssh/known_hosts*. Respóstase *yes* para establecer a conexión SSH.
kali@kaliB:~$ cat ~/.ssh/known_hosts #Observar que a conexión tipo lugar co cal gardouse o host key no ficheiro ~/.ssh/known_hosts
kali@kaliB:~$ rm -f ~/.ssh/known_hosts #Eliminar sen pedir confirmación o ficheiro ~/.ssh/known_hosts, no cal gárdanse as hosts keys dos servidores coñecidos tras conexións SSH.
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión no porto TCP por defecto 22 e co usuario kali. E como agora non se introduce por teclado a opción correspondente á directiva *StrictHostKeyChecking* actívase a configuración correspondente a esa directiva no ficheiro *~/.ssh/config*
kali@kaliB:~$ cat ~/.ssh/known_hosts #Observar que a conexión tipo lugar co cal gardouse o host key no ficheiro ~/.ssh/known_hosts
kali@kaliB:~$ rm -f ~/.ssh/known_hosts #Eliminar sen pedir confirmación o ficheiro ~/.ssh/known_hosts, no cal gárdanse as hosts keys dos servidores coñecidos tras conexións SSH.
kali@kaliB:~$ sed -i 's/Host */Host !192.168.120.100/' ~/.ssh/config #Habilitar as directivas do ficheiro **~/.ssh/config** a todos os hosts agás para o host 192.168.120.100, é dicir, agora as directivas non funcionan para o host 192.168.120.100
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión no porto TCP por defecto 22 e co usuario kali. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro *~/.ssh/config*, e neste caso para o host 192.168.120.100 como *StrictHostKeyChecking* está a *yes* entón pregúntase pola confirmación do host key do servidor gardándose para establecer a conexión e poder gardala no ficheiro *~/.ssh/known_hosts*
kali@kaliB:~$ cat ~/.ssh/known_hosts #Observar que a conexión tipo lugar co cal gardouse o host key no ficheiro ~/.ssh/known_hosts
kali@kaliB:~$ rm -f ~/.ssh/known_hosts #Eliminar sen pedir confirmación o ficheiro ~/.ssh/known_hosts, no cal gárdanse as hosts keys dos servidores coñecidos tras conexións SSH.
kali@kaliB:~$ sed -i 's/Host !192.168.120.100/Host 192.168.120.100' ~/.ssh/config #Habilitar as directivas do ficheiro **~/.ssh/config** soamente ao host 192.168.120.100, é dicir, agora as directivas soamente funcionan para o host 192.168.120.100
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión no porto TCP por defecto 22 e co usuario kali. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro

~/.ssh/config, e neste caso para o host 192.168.120.100 como *StrictHostKeyChecking* está a *no* entón non se pregunta pola confirmación do host key do servidor gardándose no ficheiro *~/.ssh/known_hosts* estableceéndose a conexión SSH.

kali@kaliB:~$ cat ~/.ssh/known_hosts #Observar que a conexión tipo lugar co cal gardouse o host key no ficheiro ~/.ssh/known_hosts

11. **ssh_config: Modificar na configuración de acceso o usuario de conexión por defecto ao servidor SSH.**

    - **User** ➜ Directiva que determina o usuario que fai login para establecer a conexión.

kali@kaliB:~$ cat > ~/.ssh/config <<EOF Comezo do ficheiro a crear ~/.ssh/config
Host * #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
User root #Se non se especifica por liña de comandos facer login co usuario root.
EOF Comezo do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ ssh -o User=kali kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión no porto TCP por defecto 22 e co usuario kali. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro *~/.ssh/config*, pero neste caso introdúcese por liña de comandos mediante a opción *-o* que o *User* toma o valor *kali*.
kali@kaliB:~$ ssh 192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, pero como non determinamos o usuario co que facemos por liña de comandos, lese o ficheiro *~/.ssh/config*, no cal o usuario para todas as conexións *(Host *)* é root *(User root)*, e o login con este usuario non está permitido, co cal non se establece a conexión SSH.
kali@kaliB:~$ sed -i 's/User root/User kali/' ~/.ssh/config #Habilitar por defecto o usuario kali como o usuario a establecer login mediante conexións SSH.
kali@kaliB:~$ ssh 192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, pero como non determinamos o usuario co que facemos por liña de comandos, lese o ficheiro *~/.ssh/config*, no cal o usuario para todas as conexións *(Host *)* é kali *(User kali)*, e o login con este usuario está permitido, co cal agora si se establece a conexión SSH.

12. **ssh_config: Modificar na configuración de acceso o porto TCP de conexión ao servidor SSH.**

- **Port** ➜ Directiva que determina o porto TCP co que establecer conexión no servidor SSH.

kali@kaliB:~$ cat > ~/.ssh/config <<EOF Comezo do ficheiro a crear ~/.ssh/config
Host * #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
Port 9999 #Se non se especifica por liña de comandos facer conexión ao porto TCP 9999 do servidor SSH.
EOF Comezo do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ ssh -o Port=1111 kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, establecendo a conexión co usuario kali. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro *~/.ssh/config*, pero neste caso introdúcese por liña de comandos mediante a opción *-o* que o *Port* toma o valor *1111*.
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, pero como non determinamos o porto TCP de conexión por liña de comandos, lese o ficheiro *~/.ssh/config*, no cal o porto TCP por defecto para as conexións SSH é o 9999, e como o servidor non está a esperar conexións neste porto non se establece a conexión SSH.
kali@kaliB:~$ sed -i 's/Port 9999/Port 22/' ~/.ssh/config #Por defecto no ficheiro estableceranse as conexións SSH accedendo ao porto TCP 22
kali@kaliB:~$ ssh kali@192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB, pero como non determinamos o porto TCP de conexión por liña de comandos, lese o ficheiro *~/.ssh/config*, no cal agora o porto TCP por defecto para as conexións SSH é o 22, e como o servidor si está a esperar conexións neste porto si se establece a conexión SSH.

13. **ssh_config: Modificar múltiples opcións na configuración de acceso de conexión ao servidor SSH.**

- **StrictHostKeyChecking ➜** Directiva que determina se se confía na key host do servidor SSH co que se establece a conexión. Os hosts keys son gardados por defecto no ficheiro ~/.ssh/known_hosts.
- **User ➜** Directiva que determina o usuario que fai login para establecer a conexión.
- **Port ➜** Directiva que determina o porto TCP co que establecer conexión no servidor SSH.

kali@kaliB:~$ cat > ~/.ssh/config <<EOF Comezo do ficheiro a crear ~/.ssh/config
Host 192.168.120.100 #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
StrictHostKeyChecking no #Se non se especifica por liña de comandos gardar automaticamente en ~/.ssh/known_hosts a host key do servidor a conectar.
User root #Se non se especifica por liña de comandos facer login co usuario root.
Port 9999 #Se non se especifica por liña de comandos facer conexión ao porto TCP 9999 do servidor SSH.
EOF Comezo do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ ssh 192.168.120.100 cat /etc/passwd #Executar o comando **cat /etc/passwd** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB. Para aquelas opcións que non se establecen por liña de comandos lese o ficheiro *~/.ssh/config*, e neste caso para o host 192.168.120.100 como:

- *StrictHostKeyChecking* está a *no* entón non se pregunta pola confirmación do host key do servidor gardándose no ficheiro *~/.ssh/known_hosts* establecéndose a conexión SSH.
- *User* é *root* co cal o usuario co que se intenta realizar a conexión é **root**, e o login con este usuario non está permitido, co cal non se establece a conexión SSH.
- *Port* toma o valor 9999, sendo este o porto TCP co se intenta realizar a conexión co servidor SSH, e como o servidor non está a esperar conexións neste porto non se establece a conexión SSH.

kali@kaliB:~$ ssh -o StrictHostKeyChecking=no -o Port=9999 -l root 192.168.120.100 #Neste comando empréganse as opcións do comando ssh equivalentes á configuración do ficheiro *~/.ssh/config* anterior, pero agora lense por liña de comandos.
kali@kaliB:~$ ssh -o StrictHostKeyChecking=no -p 9999 root@192.168.120.100 #Comando equivalente ao anterior.

14. **Verificar Preferencias: liña de comandos → ~/.ssh/config → /etc/ssh/ssh_config**

- **Port** ➜ Directiva que determina o porto TCP co que establecer conexión no servidor SSH.

kali@kaliB:~$ cat > ~/.ssh/config <<EOF Comezo do ficheiro a crear ~/.ssh/config
Host 192.168.120.100 #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
Port 9999 #Se non se especifica por liña de comandos facer conexión ao porto TCP 9999 do servidor SSH.
EOF Comezo do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ sudo bash -c "echo 'Port 8888' >> /etc/ssh/ssh_config" Empregar sudo para obter permisos para poder engadir a directiva Port ao ficheiro /etc/ssh/ssh_config
kali@kaliB:~$ ssh -p 22 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **22**, posto que as opcións de liña de comandos prevalecen sobre os ficheiros: ~/.ssh/config e /etc/ssh/ssh_config
kali@kaliB:~$ ssh 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **9999**, posto que como explicitamente non está posto o porto a acceder na liña de comandos, o ficheiro ~/.ssh/config prevalece sobre /etc/ssh/ssh_config
kali@kaliB:~$ sed -i 's/Port/#Port/' ~/.ssh/config #Comentar a directiva Port en ~/.ssh/config
kali@kaliB:~$ ssh 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **8888**, posto que como explicitamente non está posto o porto a acceder na liña de comandos, e tampouco existe activa a directiva Port no ficheiro ~/.ssh/config, intenta o acceso no porto TCP 8888 por estar activa a directiva no ficheiro /etc/ssh/ssh_config

15. **Verificar Preferencias: Posición directivas**

- **Port** ➜ Directiva que determina o porto TCP co que establecer conexión no servidor SSH.

kali@kaliB:~$ cat > ~/.ssh/config <<EOF Comezo do ficheiro a crear ~/.ssh/config
Host 192.168.120.100 #Afecta ás seguintes directivas, desta sección, ata atopar outra directiva *Hosts* ou *Match*
Port 9999 #Se non se especifica por liña de comandos facer conexión ao porto TCP 9999 do servidor SSH.
Port 1111 #Se non se especifica por liña de comandos facer conexión ao porto TCP 1111 do servidor SSH. *Como esta directiva Port aparece logo doutra directiva Port na mesma sección Host non ten efecto.*
EOF Comezo do ficheiro a crear ~/.ssh/config
kali@kaliB:~$ sudo sed -i 's/Port 8888/Port 8888\nPort 2222/' /etc/ssh/ssh_config Empregar sudo para obter permisos para poder engadir unha nova directiva Port ao ficheiro /etc/ssh/ssh_config
kali@kaliB:~$ ssh -p 22 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **22**, posto que as opcións de liña de comandos prevalecen sobre os ficheiros: ~/.ssh/config e /etc/ssh/ssh_config
kali@kaliB:~$ ssh 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **9999**, posto que como explicitamente non está posto o porto a acceder na liña de comandos, o ficheiro ~/.ssh/config prevalece sobre /etc/ssh/ssh_config. E neste ficheiro prevalece na mesma sección a primeira definición de directiva Port que apareza.
kali@kaliB:~$ sed -i 's/Port/#Port/' ~/.ssh/config #Comentar as directivas Port en ~/.ssh/config
kali@kaliB:~$ ssh 192.168.120.100 #Intento de conexión ao servidor SSH no porto TCP **8888**, posto que como explicitamente non está posto o porto a acceder na liña de comandos, e tampouco existe activa ningunha directiva Port no ficheiro ~/.ssh/config, intenta o acceso no porto TCP 8888 por ser na mesma sección a primeira directiva Port activa no ficheiro /etc/ssh/ssh_config

---

Ricardo Feijoo Costa

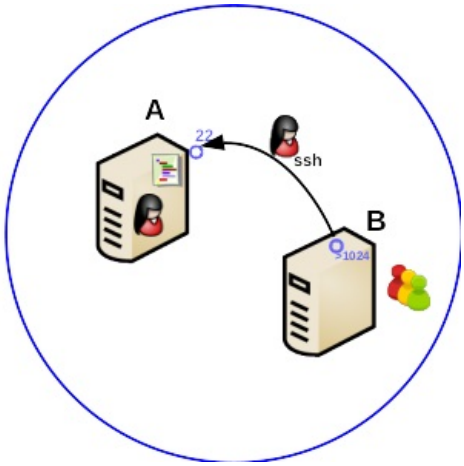
Conexión Remota mediante SSH
Cambios en sshd_config

ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede Interna
Servidor SSH: openssh-server
ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
ISO: Kali Live amd64
IP/MS: 192.168.120.101/24



NOTAS:

- **Servidor ssh GNU/Linux**:
  - Paquete openssh-server (# apt update && apt -y install openssh-server).
  - Ficheiro de configuración: **/etc/ssh/sshd_config (man sshd_config)**

    PermitRootLogin ➜ Directiva que determina se o usuario root pode acceder a conectarse mediante SSH.

    Port ➜ Directiva que determina o porto TCP de escoita para o servidor SSH. Poden existir múltiples liñas Port indicando diferentes portos de escoita do servizo SSH.

    ListenAddress ➜ Directiva que determina que direccións locais escoitan no servizo SSH. Poden existir múltiples liñas ListenAddress indicando varias interfaces de escoita do servizo SSH.

    X11Forwarding ➜ Directiva que determina se a redirección gráfica é posible mediante conexións SSH. Pode soamente tomar 2 valores: yes/no.

    X11DisplayOffset ➜ Directiva que determina o número do display onde espera o servidor gráfico. Por defecto é 10, para evitar interferencias con reais servidores X11.

    X11UseLocalhost ➜ Directiva que determina se a redirección gráfica é posible na dirección loopback ou en calquera dirección.

    PubKeyAuthentication ➜ Directiva que determina se a autenticación de cifrado asimétrico (ou cifrado de clave pública) está permitida. Por defecto está permitida tomando o valor *yes*

    AuthorizedKeysFile ➜ Directiva que especifica o ficheiro que contén as claves públicas empregadas para a autenticación de usuario. Por defecto o ficheiro toma o valor *~/.ssh/authorized_keys* ou *~/.ssh/authorized_keys2*

    PermitEmptyPasswords ➜ Directiva que especifica se a autenticación de usuario é posible con contrasinais baleiras. Por defecto, toma o valor *no* impedindo o acceso cunha contrasinal baleira.

    PasswordAuthentication ➜ Directiva que especifica se a autenticación de usuario é posible mediante contrasinal. Por defecto, toma o valor *yes* permitindo o acceso a través de contrasinal. Soe configurarse a *no* cando soamente interesa acceder mediante cifrado asimétrico.

    MaxAuthTries ➜ Directiva que especifica o número máximo de reintentos de autenticación por conexión. Por defecto son *6*

NOTAS:

- **Servidor gráfico X (Xorg)**: Variable de contorna **DISPLAY (man X ; man ssh)**

  Un **display** consta mínimo de 3 elementos: teclado, rato e pantalla. Dende a perspectiva dun usuario todo servidor gráfico ten un display, o cal defínese como ➜ hostname:displaynumber.screennumber

  DISPLAY=hostname:displaynumber.screennumber ➜ Variable de contorna que permite definir o display dun servidor gráfico:

  - hostname: Especifica o nome do host no cal o display está fisicamente conectado. Se non se define enténdese que a comunicación ao servidor gráfico sobre a mesma máquina (maquina local) terá lugar da forma máis eficiente.
  - displaynumber: É o único valor do DISPLAY que sempre debe configurarse. O termo display é usado normalmente para referirse a un conxunto de monitores que comparten un conxunto común de dispositivos de entrada (teclado, rato, tablet, etc.). A maioría dos equipos soamente posúen un display, pero pode ser que posúan varíos. xa que é necesario que varios usuarios poidan traballar nunha contorna gráfica simultaneamente. Para evitar confusión, cada display sobre un equipo ten asignado un número de display (comenzando en 0) cando o servidor gráfico (X) arranca para ese display. **O número do display sempre debe darse para configurar o DISPLAY**
  - screennumber: Algúns displays comparten os seus dispositivos de entrada con 2 ou máis monitores, que poden ser configurados como unha única pantalla lóxica, o al permite ás ventás moverse entre as pantallas, ou como pantallas individuais. Se se configura como que cada monitor ten as súas propias ventás, cada pantalla é asignada a un screen number. Se non se especifica o screen number toma o valor 0

  DISPLAY=:0.0 ➜ Display por defecto, que equivale a hostname=comunicación máis eficiente co servidor gráfico na mesma máquina local, displaynumber=0 e screennumber=0

  ```
  $ declare -p | grep DISPLAY
  declare -x DISPLAY=":0.0" #Amosa o valor da variable DISPLAY e como está declarada
  ```
  . Vemos que a variable está exportada, de tal xeito que a variable é válida na contorna actual da shell e en calquera subshell.

  Ben, pero que acontece cando non conectamos dende a máquina local senón dende a rede? Pois necesitamos definir o DISPLAY para que poida ser empregado dende a rede. Por exemplo, exportando a variable co nome do noso hostname. Así, se o hostname posúe o valor kaliA deberiamos facer o seguinte:

  ```
  $ declare -x DISPLAY=kaliA:0
  ```

  Nas conexións SSH se empregamos o comando ssh coa opción -X podemos executar un programa no servidor gráfico remoto e visualizalo no equipo cliente. Neste caso, automaticamente xa o comando ssh configura a variable DISPLAY correctamente para poder facer a redirección gráfica.

  A maioría dos programas aceptan na liña de comandos a opción **-display DISPLAY** ou **--display DISPLAY**, a cal **sobreescribe** o contido da variable **DISPLAY**:

  ```
  $ xeyes -display :0.0 & #Executar en segundo plano o comando xeyes no display :0.0 (hostname=servidor gráfico local, display=0, screennumber=0)
  ```

  Podemos arrancar outro sistema X Window (servidor gráfico + cliente gráfico) mediante o comando **startx (man startx; man xinit)**:

  ```
  $ su - -c "startx -- :10" #No comando startx antes dos caracteres -- teñen lugar as opcións do cliente gráfico, e logo teñen lugar as opcións do servidor gráfico. Así, creamos o display :10 (hostname=servidor gráfico local, display=10, screennumber=0). O comando é executado mediante o usuario root a través de su - -c
  $ su - -c "xeyes -display :10" #Executar coma root o comando xeyes no display :10 (hostname=servidor gráfico local, display=10, screennumber=0)
  ```

Práctica - Conexión Remota mediante SSH - Cambios en sshd_config

Máquina virtual A: Kali amd64

1. Pór contrasinal ao usuario, co que arranca a live, de nome: kali
   Na contorna gráfica abrir un terminal e executar:

   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali || (echo -e 'kali\nabc123.\nabc123.' | passwd) #Cambiar o contrasinal do usuario kali. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final).
   kali@kali:~$ sudo passwd root || (sudo -c "echo -e 'abc123.\nabc123.' | passwd") #Cambiar o contrasinal do usuario root. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final). O cambio de contrasinal é posible debido aos permisos configurados co comando sudo (/etc/sudoers, visudo).

2. Cambiar hostname da máquina virtual A. Por kaliA como hostname:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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:~$ 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:~#

5. sshd_config: Impedir/Permitir a root a conexión mediante SSH

PermitRootLogin ➜ Directiva que determina se o usuario root pode acceder a conectarse mediante SSH. Así, pode tomar os seguintes valores:

- PermitRootLogin prohibit-password ➜ o usuario root non poderá conectar mediante SSH.
- PermitRootLogin without-password ➜ igual que prohibit-password, pero está obsoleta (en desuso).
- PermitRootLogin yes ➜ o usuario root poderá conectar mediante SSH.
- PermitRootLogin forced-commands-only ➜ o usuario root poderá conectar realizar conexións de comandos (non consola) SSH mediante cifrado asimétrico (cifrado de clave pública).
- PermitRootLogin no ➜ deshabilita ao usuario root o acceso mediante conexión SSH.

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# grep -n '^#PermitRootLogin' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'#PermitRootLogin'* nun comezo de liña *^* indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^#PermitRootLogin' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/PermitRootLogin/#PermitRootLogin/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *PermitRootLogin* por *#PermitRootLogin* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *PermitRootLogin*

root@kaliA:~# sed -i "${VAR}a\PermitRootLogin yes" /etc/ssh/sshd_config #Activar o acceso a root mediante conexión SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *PermitRootLogin yes* logo da liña atopada anteriormente que comeza por *#PermitRootLogin*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -v root@localhost #Realizar unha conexión SSH a localhost mediante o usuario root 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.

root@kaliA:~# exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# sed -i 's/PermitRootLogin yes/PermitRootLogin prohibit-password/' /etc/ssh/sshd_config #Deshabilitar conexións SSH ao usuario root.

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -v root@localhost #Realizar unha conexión SSH a localhost mediante o usuario root 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.
Neste caso como modificamos a directiva *PermitRootLogin* a *prohibit-password* o acceso co usuario *root* non é posible.

root@kaliA:~# exit #Saír da consola local do usuario **root** para voltar á consola local do usuario **kali**.

6. sshd_config: Modificar o porto TCP de conexión SSH

Port ➜ Directiva que determina o porto TCP de escoita para o servidor SSH. Poden existir múltiples liñas Port indicando diferentes portos de escoita do servizo SSH:

- Port 22 ➜ por defecto o servidor SSH espera no porto TPC 22 (ver /etc/services).
- Port 4444 ➜ modificación do porto de escoita do servidor SSH ao porto TCP 4444.

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# grep -n '^#Port' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'#Port'* nun comezo de liña ^ indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^#Port' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/Port/#Port/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *Port* por *#Port* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *Port*

root@kaliA:~# sed -i "${VAR}a\Port 4444" /etc/ssh/sshd_config #Activar o porto TCP de escoita 4444 para esperar conexións SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *Port 4444* logo da liña atopada anteriormente que comeza por *#Port*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -p 4444 kali@localhost #Realizar unha conexión SSH a localhost mediante o usuario kali e o seu contrasinal a través do porto TCP 4444. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# sed -i "${VAR}a\Port 22" /etc/ssh/sshd_config #Activar o porto TCP de escoita 22 para esperar conexións SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *Port 22* logo da liña atopada anteriormente que comeza por *#Port*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -p 22 kali@localhost #Realizar unha conexión SSH a localhost mediante o usuario kali e o seu contrasinal a través do porto TCP 22. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

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 do usuario **root** para voltar á consola local do usuario **kali**.

7. sshd_config: Especificar as interfaces de rede de escoita para o servizo SSH

ListenAddress ➜ Directiva que determina que direccións locais escoitan no servizo SSH. Poden existir múltiples liñas ListenAddress indicando varias interfaces de escoita do servizo SSH. Pódese empregar os seguintes formatos:

- ListenAddress 0.0.0.0 ➜ é a opción por defecto. O servizo SSH está a escoita de conexións en todas as interfaces de rede locais.
- ListenAddress localhost ➜ indica o hostname onde espera a escoita o servizo SSH, neste caso localhost (ver /etc/hosts)
- ListenAddress kaliA ➜ indica o hostname onde espera a escoita o servizo SSH, neste caso kaliA (ver /etc/hosts)
- ListenAddress kaliA:5555 ➜ indica o hostname e o porto TCP onde espera á escoita o servizo SSH, neste caso no host kaliA (ver /etc/hosts) e no porto TCP 5555
- ListenAddress 127.0.0.1 ➜ indica a IP onde espera o a escoita o servizo SSH, neste caso 127.0.0.1
- ListenAddress 127.0.0.1:6666 ➜ indica a IP e o porto TCP onde espera á escoita o servizo SSH, neste caso 127.0.0.1 e no porto TCP 6666

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# grep -n '^#ListenAddress' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'#ListenAddress'* nun comezo de liña ^ indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^#ListenAddress' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/ListenAddress/#ListenAddress/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *ListenAddress* por *#ListenAddress* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *ListenAddress*

root@kaliA:~# sed -i "${VAR}a\ListenAddress 127.0.0.1" /etc/ssh/sshd_config #Pór soamente (xa que comentamos anteriormente todas as directivas ListenAddress) á escoita a IP interface loopback(lo) 127.0.0.1.

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -l kali 127.0.0.1 #Intento de conexión SSH a través do porto TCP 22 en 127.0.0.1 mediante o 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.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# ssh kali@localhost #Intento de conexión SSH a través do porto TCP 22 en localhost (/etc/hosts) mediante o 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.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# sed -i "${VAR}a\ListenAddress localhost:6666" /etc/ssh/sshd_config #Pór á escoita o host localhost (interface loopback(lo) 127.0.0.1) no porto TCP 6666.

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -p 6666 kali@localhost #Intento de conexión SSH a través do porto TCP 6666 en localhost mediante o 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.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# sed -i "${VAR}a\ListenAddress 192.168.120.100:7777" /etc/ssh/sshd_config #Pór á escoita a interface configurada coa IP 192.168.120.100 no porto TCP 7777.

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -p 7777 kali@localhost #Intento de conexión SSH a través do porto TCP 7777 en localhost mediante o usuario kali e o seu contrasinal. Pero como localhost non está configurado para acceder por ese porto non é posible a conexión.

root@kaliA:~# ssh -p 7777 -l kali 192.168.120.100 #Intento de conexión SSH a través do porto TCP 7777 en 192.168.120.100 mediante o 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.

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 do usuario **root** para voltar á consola local do usuario **kali**.

8. **sshd_config: Redireccionar X (X11Forwarding, X11DisplayOffset, X11UseLocalhost) para realizar conexións gráficas remotas en conexións SSH**

X11Forwarding ➜ Directiva que determina se a redirección gráfica é posible mediante conexións SSH. Pode soamente tomar 2 valores: yes/no.

- X11Forwarding no ➜ é o valor por defecto. Deshabilita a redirección gráfica do servidor SSH.
- X11Forwarding yes ➜ Habilita a redirección gráfica do servidor SSH.

X11DisplayOffset ➜ Directiva que determina o número do display onde espera o servidor gráfico. Por defecto é 10, para evitar interferencias con reais servidores X11.

- X11DisplayOffset 10 ➜ é o valor por defecto. Indica o número de display onde espera o servidor gráfico para conexións SSH.
- X11DisplayOffset 100 ➜ Indica o número de display a 100 onde espera o servidor gráfico para conexións SSH.
- X11Forwarding yes ➜ Habilita a redirección gráfica do servidor SSH.

X11UseLocalhost ➜ Directiva que determina se a redirección gráfica é posible na dirección loopback ou en calquera dirección:

- X11UseLocalhost yes ➜ é o valor por defecto. Permite a redirección gráfica á dirección loopback e define a variable de entorno DISPLAY a localhost, o cal prevén conexións remotas non permitidas ao display.
- X11UseLocalhost no ➜ Habilita a redirección gráfica a todas as interfaces de rede.

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# grep -n '^X11Forwarding' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'X11Forwarding'* nun comezo de liña ^ indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^X11Forwarding' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/X11Forwarding/#X11Forwarding/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *X11Forwarding* por *#X11Forwarding* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *X11Forwarding*

root@kaliA:~# sed -i "${VAR}a\X11Forwarding yes" /etc/ssh/sshd_config #Activar o acceso a root mediante conexión SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *X11Forwarding yes* logo da liña atopada anteriormente que comeza por *#X11Forwarding*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -X kali@localhost #Realizar unha conexión SSH a localhost mediante o usuario kali e o seu contrasinal a través do porto TCP 22 solicitando a redirección gráfica (*-X*). Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

root@kaliA:~# exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# grep -n '^#X11DisplayOffset' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'#X11DisplayOffset'* nun comezo de liña ^ indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^#X11DisplayOffset' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/X11DisplayOffset/#X11DisplayOffset/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *X11DisplayOffset* por *#X11DisplayOffset* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *X11DisplayOffset*

root@kaliA:~# sed -i "${VAR}a\X11DisplayOffset 100" /etc/ssh/sshd_config #Redirección gráfica no display 100 en conexións SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *X11DisplayOffset 100* logo da liña atopada anteriormente que comeza por *#X11DisplayOffset*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# ssh -X kali@localhost #Realizar unha conexión SSH a localhost mediante o usuario kali e o seu contrasinal a través do porto TCP 22 solicitando a redirección gráfica (*-X*), agora no display 100. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# grep -n '^#X11UseLocalhost' /etc/ssh/sshd_config #Buscar no ficheiro /etc/ssh/sshd_config mediante o comando *grep* o patrón de texto *'#X11UseLocalhost'* nun comezo de liña ^ indicando o número/s da/s liña/s (*-n*) onde existe o patrón buscado.

root@kaliA:~# VAR=$(grep -n '^#X11UseLocalhost' /etc/ssh/sshd_config | cut -d ':' -f1 | tail -1) #Crear a variable VAR co contido o número da última liña (poden existir máis de 1 se o ficheiro xa fora modificado) do patrón buscado no ficheiro /etc/ssh/sshd_config

root@kaliA:~# sed -i "s/X11UseLocalhost/#X11UseLocalhost/g" /etc/ssh/sshd_config #Modificar no ficheiro /etc/ssh/sshd_config o patrón de texto *X11UseLocalhost* por *#X11UseLocalhost* en todas as concurrencias atopadas, inclusive que fosen na mesma liña, é dicir, comentar (mediante o caracter *#*) para que non teña lugar a directiva *X11UseLocalhost*

root@kaliA:~# sed -i "${VAR}a\X11UseLocalhost yes" /etc/ssh/sshd_config #Redirección gráfica onde soamente está permitida a interface loopback para exportación gráfica en conexións SSH, é dicir, modificar o ficheiro /etc/ssh/sshd_config engadindo a liña *X11UseLocalhost yes* logo da liña atopada anteriormente que comeza por *#X11UseLocalhost*

root@kaliA:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

kali@kaliA:~$ ssh -X kali@localhost #Realizar unha conexión SSH a localhost mediante o usuario kali e o seu contrasinal a través do porto TCP 22 solicitando a redirección gráfica (*-X*), permitindo soamente a exportación gráfica a través de localhost. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

kali@kaliA:~$ firefox & #Lanzar o navegador firefox, realizando a execución en segundo plano (&). Agora é posible mediante SSH visualizar comandos que empreguen o servidor gráfico debido a que temos activada a redirección gráfica.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

Máquina virtual B: Kali amd64

NOTAS:

- **Cliente ssh GNU/Linux:**
  - Comando ssh. Paquete openssh-client (# apt update && apt -y install openssh-client).
  - Ficheiro de configuración: **/etc/ssh/ssh_config (man ssh_config)**

9. Pór contrasinal ao usuario, co que arranca a live, de nome: kali
   Na contorna gráfica abrir un terminal e executar:

   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ passwd kali || (echo -e 'kali\nkaliBpass\nkaliBpass' | passwd) #Cambiar o contrasinal do usuario kali. Por como contrasinal **kaliBpass**

10. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

    OPCIÓN A:
    kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
    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**.

    OPCIÓN B:
    kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
    root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
    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**.

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

    kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
    root@kaliB:~# /etc/init.d/avahi-daemon stop || systemctl stop avahi-daemon #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@kaliB:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
    root@kaliB:~# 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@kaliB:~# 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@kaliB:~# 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@kaliB:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
    root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa Máquina Virtual A na IP 192.168.120.100
    root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

12. **Comprobar estado do Servidor SSH e execución de comandos a través da conexión SSH:**

kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh (192.168.120.100) 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 kali@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el en 192.168.120.100 co usuario kali e o seu contrasinal no porto TCP 22. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter.

kali@kaliA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

kali@kaliB:~$ ssh kali@192.168.120.100 "netstat -natp | grep 22" #Executar o comando **netstat** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB.

kali@kaliB:~$ ssh kali@192.168.120.100 ss -natp #Executar o comando **ss** no servidor SSH 192.168.120.100 e ver a saída da execución do comando na máquina local kaliB.

kali@kaliB:~$ ssh -X kali@192.168.120.100 "thunar &" #Lanzar en segundo plano o explorador de arquivos thunar da máquina 192.168.120.100 a través dunha conexión SSH e como está activada a redirección gráfica é posible visualizar na máquina local (kaliB) o administrador de arquivos da máquina virtual A (kaliA).

kali@kaliB:~$ ssh -X kali@192.168.120.100 "firefox &" #Lanzar en segundo plano o navegador firefox da máquina 192.168.120.100 a través dunha conexión SSH e como está activada a redirección gráfica é posible visualizar na máquina local (kaliB) o navegador gráfico da máquina virtual A (kaliA).

Ricardo Feijoo Costa

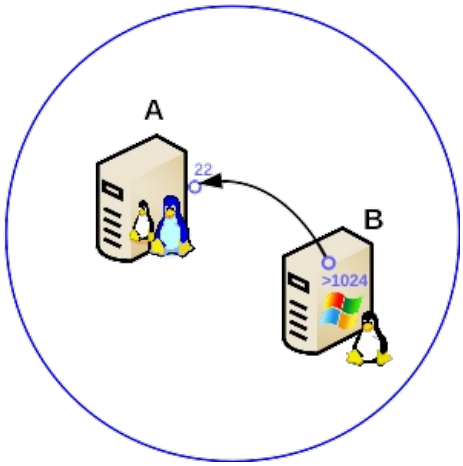
Cifrado asimétrico
Conexión Remota mediante SSH sen contrasinal

ESCENARIO

Máquinas virtuais ou físicas:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina A:
Servidor SSH
IP/MS: 192.168.120.100/24
SO: Kali Live amd64

Máquina B:
Cliente SSH
IP/MS: 192.168.120.101/24
SO_1: Live GNU/Linux
SO_2: Microsoft Windows



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:

- Cliente ssh GNU/Linux: **comando ssh (paquete openssh-client)**
- Cliente ssh Microsoft Windows: **putty**
- Documentación sobre **putty**

Práctica Cifrado asimétrico - Conexión Remota mediante SSH sen contrasinal

Arrancar coa Kali Live amd64

1. Pór contrasinal ao usuario, co que arranca a live, de nome: kali
   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. Configurar a rede:
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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 A, as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# 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@kali:~# 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).

- Se a interface eth0 non está UP, é dicir, está en estado DOWN, executar:
  root@kali:~# ip link set up dev eth0 && ip addr show eth0

root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0

3. Comprobar estado do Servidor SSH:

root@kali:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.
root@kali:~# 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@kali:~# 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@kali:~# 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@kali:~# 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@kali:~# /etc/init.d/ssh start #Arrancar o servidor SSH.
root@kali:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.
root@kali:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
root@kali:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)
root@kali:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
root@kali:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled
root@kali:~# 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@kali:~# 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.

4. Configurar cifrado asimétrico:

kali@kali:~$ ssh-keygen -t rsa #Crear un par de chaves: pública e privada. No comando emprégase o algoritmo de cifrado rsa (Rivest, Shamir y Adleman), que por defecto a non ser que o modifiquemos co parámetro -b nº_bits é de 2048bits.

- Debemos elixir o cartafol onde gardar as chaves e o nome destas. Pulsamos Enter para deixar por defecto o cartafol .ssh/ e o nome id_rsa dentro do HOME do usuario: /home/kali.
- Passphrase nulo. Se aquí pomos un contrasinal, frase ou similar, cando queiramos conectarnos ao Servidor SSH en vez de pedir o contrasinal do usuario da conexión pedirá iste passphrase, mais como cando queremos conectarnos queremos facelo de forma directa sen petición de contrasinal ou passphrase, entón pulsamos 2 veces Enter para que a conexión se faga sen contrasinal.
- Chave pública e privada creadas. Fingerprint. Creáronse no cartafol anteriormente indicado a chave privada id_rsa e a chave pública id_rsa.pub. Tamén creouse o fingerprint da chave pública, e dicir, a identificación inequívoca da chave pública correspondente ao usuario kali deste equipo.

kali@kali:~$ ls -lahtr $HOME/.ssh #Executar o comando ls dentro do cartafol de traballo do usuario ($HOME=/home/kali) coas opcións -l, -a, -h, -t e -r. A opción -l permite amosar de forma extendida o atopado (tipo de ficheiro, permisos, propietarios...), a opción -h engade unha letra indicativa de tamaño, tal como M para megabytes binarios (`mebibytes'), a cada tamaño. A opción -t clasifica polo tempo de modificación (o `mtime' no inodo) en vez de alfabeticamente, cos ficheros máis recientes en primeiro lugar. A opción -r clasifica en orde inversa. Polo tanto, o comando lista ficheiros e directorios do directorio /home/kali amosando de abaixo hacia arriba os máis recentes e en formato de lectura de tamaño máis amigable para as persoas (K, M, G...)

De interese: Comprobar os **permisos** dos ficheiros: **id_rsa**, **id_rsa.pub, authorized_keys**

kali@kali:~$ ssh-copy-id -i .ssh/id_rsa.pub kali@localhost #Copia da chave pública ao Servidor SSH. Para poder establecer a conexión sen contrasinal enviamos unha copia da chave pública ao Servidor SSH. Soamente será posible establecer unha conexión sen contrasinal se posuimos a parella desa chave pública, que non é outra que a chave privada, polo cal, nunca deberiamos desprendernos da chave privada, xa que sen ela a conexión non sería posible ou outro usuario podería suplantarnos no caso de facerse coa chave privada.

- Password usuario kali: Como aínda non temos copiada a chave pública nesta conexión pídese o contrasinal do usuario co cal queremos conectarnos ao Servidor SSH: kali. A password do usuario kali é **abc123.** (Ollo que o contrasinal ten un caracter punto final)
- Agora a conexión sen contrasinal será posible para o usuario kali, con todos os permisos deste usuario, na máquina Servidor SSH (localhost).

kali@kali:~$ ls -lahtr $HOME/.ssh #Executar o comando ls dentro do cartafol de traballo do usuario ($HOME=/home/kali) coas opcións -l, -a, -h, -t e -r. A opción -l permite amosar de forma extendida o atopado (tipo de ficheiro, permisos, propietarios...), a opción -h engade unha letra indicativa de tamaño, tal como M para megabytes binarios (`mebibytes'), a cada tamaño. A opción -t clasifica polo tempo de modificación (o `mtime' no inodo) en vez de alfabeticamente, cos ficheros máis recientes en primeiro lugar. A opción -r clasifica en orde inversa. Polo tanto, o comando lista ficheiros e directorios do directorio /home/kali amosando de abaixo hacia arriba os máis recentes e en formato de lectura de tamaño máis amigable para as persoas (K, M, G...)

De interese: Comprobar os **permisos** dos ficheiros: **id_rsa**, **id_rsa.pub, authorized_keys**

kali@kali:~$ ssh -v kali@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost sen contrasinal, a través de cifrado asimétrico. 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@kali:~$ exit #Saír da consola remota ssh a que acabamos de acceder mediante cifrado asimétrico, para voltar á consola local de **kali**.

kali@kali:~$ exit #Saír da outra consola remota ssh que accederamos mediante contrasinal, para voltar á consola local de **root**.

root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

kali@kali:~$

Comprobar acceso ssh sen contrasinal

5. Comprobar o acceso ssh sen contrasinal dende a máquina B arrancada cunha distro Live GNU/Linux
6. Comprobar o acceso ssh sen contrasinal dende a máquina B arrancada cun Sistema Operativo Microsoft Windows

---

Servizo Web: Apache

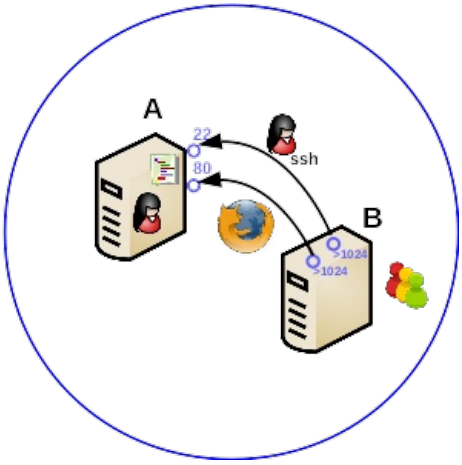
ESCENARIO

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

Máquina virtual A:
Rede Interna
Servidor SSH: openssh-server
Servidor Web: Apache (apache2)
ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
Cliente Web: Navegador (firefox)
ISO: Kali Live amd64
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:

- Cliente ssh GNU/Linux: **comando ssh (paquete 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)**
- Servidor Web Apache:
  - Paquete apache2 (# apt update && apt -y install apache2).
  - **Nomenclatura versións: 2.X.revision**, onde:
    - X toma valor par ➜ a versión é estable
    - X toma valor impar ➜ a versión é de desenvolvemento

NOTAS:

- Documentación oficial sobre o Servidor web **Apache (v2.4)**

  - Paquete apache2 (# apt update && apt -y install apache2).
  - Configuración en: **/etc/apache2/ (man apache2)**

  apache2.conf ➜ Ficheiro de configuración principal. Non deberiase modificarse con novas directivas. Así:

    - Se se quere extender a configuración global de Apache deberianse incluír noutros ficheiros de configuración dentro do directorio /etc/apache2/conf-available e activalos en /etc/apache2/conf-enabled (a2enconf, a2disconf).
    - No caso de cambiar os portos e socket de conexión deberiase modificar o ficheiro /etc/apache2/ports.conf

  Aparecen por liña pares de valores: directiva e argumento. As directivas non distinguen entre maiúsculas e minúsculas e os argumentos si poden distinguir.

  ports.conf ➜ Ficheiro de configuración que contén as directivas de configuración de portos TCP e direccións IP onde escoitar as conexións do servidor web Apache.

  envvars ➜ Ficheiro de configuración que contén as variables de entorno que poden ser empregadas na configuración polos ficheiros de configuración que o consideren.

    - APACHE_RUN_USER=www-data ➜ usuario que executa o servizo web Apache
    - APACHE_RUN_GROUP=www-data ➜ grupo que executa o servizo web Apache
    - APACHE_PID_FILE=/var/run/apache2$SUFFIX/apache2.pid ➜ pid de execución do servizo web Apache
    - APACHE_ULIMIT_MAX_FILES='ulimit -n 65536' ➜ ulimit determina o número máximo de ficheiros abertos permitidos para servizo web Apache

  conf-available ➜ Os ficheiros engadidos neste directorio son invocados (include) polo ficheiro de configuración global /etc/conf/apache2.conf. Este directorio é un bo lugar para engadir directivas na configuración. Todos os arquivos neste directorio para ser tidos en conta deben rematar coa extensión *.conf*
  Os ficheiros situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2enconf** e **a2disconf** respectivamente.

  conf-enabled ➜ Habilita a configuración dos ficheiros. Os ficheiros de configuración habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/conf-available/

    - **a2enconf ➜** Comando que permite habilitar os ficheiros de configuración situados no directorio /etc/apache2/conf-available/ engadindo ligazóns dende /etc/apache2/conf-enabled a /etc/apache2/conf-available
    - **a2disconf ➜** Comando que permite deshabilitar os ficheiros de configuración activados no directorio /etc/apache2/conf-enabled eliminando ligazóns existentes dende /etc/apache2/conf-enabled a /etc/apache2/conf-available

  mods-available ➜ Este directorio conten ficheiros que rematan coa extensión .load e .conf. Os ficheiros .load conteñen as directivas de configuración necesarias para cargar o módulo en cuestión. Os ficheiros .conf conteñen as directivas necesarias para empregar o módulo en cuestión. Os módulos situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2enmod** e **a2dismod** respectivamente.

  mods-enabled ➜ Habilita os módulos. Os módulos habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/mods-available/

    - **a2enmod ➜** Comando que permite habilitar os ficheiros dos módulos situados no directorio /etc/apache2/mods-available/ engadindo ligazóns dende /etc/apache2/mods-enabled a /etc/apache2/mods-available
    - **a2dismod ➜** Comando que permite deshabilitar os ficheiros dos módulos activados no directorio /etc/apache2/mods-enabled eliminando ligazóns existentes dende /etc/apache2/mods-enabled a /etc/apache2/mods-available

sites-available ➜ Similar ao directorio mods-available, agás que contén as directivas para diferentes VirtualHost. O hostname non ten porque corresponder exactamente co nome do ficheiro. Debian por defecto posúe 2 ficheiros neste directorio:

- 000-default.conf ➜ VirtualHost para o porto TCP 80(hhtp)(Ver arquivo /etc/services)
- default-ssl.conf ➜ VirtualHost para o porto TCP 443(https)(Ver arquivo /etc/services). Este sitio pode estar activado pero será funcional cando o modulo ssl tamén está activado.

Os sitios(arquivos) situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2ensite** e **a2dissite** respectivamente.

sites-enabled ➜ Habilita os VirtualHost(sitios). Os sitios(arquivos) habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/sites-available/

- **a2ensite ➜** Comando que permite habilitar os sitios(VirtualHost) situados no directorio /etc/apache2/sites-available/ engadindo ligazóns dende /etc/apache2/sites-enabled a /etc/apache2/sites-available
- **a2dissite ➜** Comando que permite deshabilitar os sitios(VirtualHost) activados no directorio /etc/apache2/sites-enabled eliminando ligazóns existentes dende /etc/apache2/sites-enabled a /etc/apache2/sites-available

- **Directivas**:

Include ➜ Directiva que permite engadir ficheiros de configuración á configuración global. Esta directiva ignora ficheiros que non finalizan na extensión .conf.

Require ➜ Directiva para permitir ou denegar acceso aos recursos.

DocumentRoot ➜ Directiva que define a ruta do cartafol onde o servidor web Apache aloxa as páxinas. Por exemplo: /var/www/html

Listen ➜ Directiva que define IP/Porto TCP/Protocolo onde escoita o servidor web Apache (Listen 192.168.120.100:8443 https) .

VirtualHost ➜ Directiva que define a posibilidade de aloxar varios dominios no mesmo servidor Apache.

ServerName ➜ Directiva que define o nome DNS do sitio(dominio) aloxado.

ServerAlias ➜ Directiva que define outros nomes DNS para a mesma páxina.

AllowOverride ➜ Directiva que especifica que outras directivas poden ser postas en cada ficheiro .htaccess (ficheiros de configuración por directorio).

ServerSignature ➜ Directiva que permite a configuración dunha liña de pé de páxina baixo documentos xerados polo servidor (mensaxes de erro, versión...)

Options ±argumento ➜ Directiva que controla que funcións do servidor están dispoñibles nun directorio particular. Por exemplo: Options +Indexes especifica que se non existe nun directorio o ficheiro index.html amose o contido do directorio.

Timeout ➜ Directiva que especifica a cantidade de tempo que o servidor agardará por certos eventos antes de fallar unha solicitude.

MaxKeepAliveRequests ➜ Directiva que especifica o número máximo de peticións permitidas por cada conexión persistente.

KeepAliveTimeout ➜ Directiva que especifica a cantidade de tempo que o servidor esperará solicitudes posteriores nunha conexión persistente.

ErrorLog ➜ Directiva que especifica o nome do ficheiro no que o servidor rexistrará os erros que atope. Se a ruta do ficheiro non é absoluta, suponse que é relativo ao ServerRoot.

LogFormat ➜ Directiva que especifica o formato a empregar para un ficheiro de rexistro.

ServerRoot ➜ Directiva que especifica o directorio base da instalación do servidor.

- **Seccións:**

Prioridade/Alcance	Sección	Aplicación	Exemplo
- ↓ +	Directory .htaccess (AllowOverride) ↑	FileSystem	<Directory /var/www/html/prioridade> (ruta absoluta do sistema de ficheiros ↑)
	Files		
+ ↓ -	Location	DocumentRoot	<Location /prioridade> (ruta relativa ao DocumentRoot ↑)
	VirtualHost		

Táboa. Seccións

Contexto das directivas:

Estrutura dunha sección:

```
<SectionName [pathDirectorio | "pattern"]>
    Directiva1 valor1
    Directiva2 valor2
    Directiva3 valor3
    ...
    DirectivaN valorN
</SectionName >
```

Directory ➜ Sección que inclúe directivas que actúan sobre un directorio. O directorio en cuestión sempre debe ser especificado mediante unha ruta absoluta do sistema de ficheiros.

Xeralmente, so se deberían empregar ficheiros .htaccess cando non se ten acceso ao ficheiro principal de configuración do servidor, por exemplo en servidores compartidos onde non se ten acceso como root no servidor.

Files ➜ Sección que inclúe directivas que actúan sobre os ficheiros que se especifiquen.

Location ➜ Sección similar á sección Directory. O directorio en cuestión sempre debe ser especificado mediante unha ruta relativa ao DocumentRoot.

VirtualHost ➜ Sección que inclúe directivas que actúan soamente sobre un específico VirtualHost (hostname ou IP).

Resumo Prácticas Exemplos

- No **Exemplo1. Modificar Listen**, veremos como poder escoitar en distintos portos e IPs o protocolo HTTP e o protocolo HTTPS.

- No **Exemplo2. Aloxar cartafoles**, veremos como poder aloxar múltiples páxinas web no servidor web **Apache**, pero todas pertencentes ao mesmo sitio/dominio, é dicir, todas pertencentes a exemplo.local

- No **Exemplo3. Xerar virtualhost** veremos como poder aloxar páxinas de distintos dominios no mesmo servidor web mediante a configuración de hosts virtuais ou virtualhosts.

  Os virtualhosts basicamente o que fan é permitir que un mesmo servidor web poida aloxar múltiples dominios, así configurando hosts virtuais podemos aloxar: exemplo1.local, exemplo2.local..., exemploN.local no mesmo servidor web. Cada empresa terá o seu virtualhost único e independente das demais.

  Aínda que como se comentou anteriormente cada virtualhost é único e independente dos demais, todo aquilo que non estea incluído na definición de cada virtualhost herdarase da configuración principal: /etc/apache2/apache2.conf, así, se se quere definir unha directiva común en tódolos virtualhost non se debe modificar cada un dos virtualhost introducindo esa directiva senón que se debe definir esa directiva nun arquivo de configuración dentro de /etc/apache2/conf-available e empregar o comando a2enconf para habilitar esa configuración no servidor web Apache, de tal forma que todos os virtualhost herdarán esa directiva. Por exemplo en /etc/apache2/conf-available/security.conf pódese atopala directiva ServerSignature On, que engade unha liña contendo a versión do servidor e o nome do VirtualHost.

  Existe tres tipos de virtualhost: baseados en nome, baseados en IP e baseados en varios servidores principais. Imos centrarnos nos virtualhost baseados en nome.

- No **Exemplo4. Control de acceso** imos tratar distintos tipos de control de acceso (autentificación http basic, IP), os arquivos tipo **.htaccess** e seccións <Directory> e directivas Order, Allow, Deny (todavía funcionais pero desanconsellables) e Require (<RequireAll>)

  HTTP proporciona un método de autenticación básico de usuarios: basic. Este método ante unha petición do cliente(navegador web) ao servidor cando se solicita unha URL amosará un diálogo pedindo usuario e contrasinal. Unha vez autenticado o usuario, o cliente volverá facer a petición ao servidor pero agora enviando o usuario e contrasinal, en texto claro (sen cifrar) proporcionados no diálogo. É recomendable entón se se emprega este método que se faga combinado con conexión SSL (HTTPS).

- No **Exemplo5. Prioridade seccións/directivas** imos ver que sección prevalece cando unha mesma directiva é configurada en distintas seccións (Ver Táboa. Seccións)

Servizo Web - Apache

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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) 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:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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. Modificar directiva Listen.**

Na instalación defínese en ports.conf a directiva Listen nos portos TCP 80 e 443(ver /etc/services), atendendo a calquera IP do servidor. Imos configurar esta directiva para que atenda:

a. **Listen 80 ➜** Todas as interfaces do servidor no porto TCP 80 (Configuración por defecto):

root@kaliA:~# grep Listen /etc/apache2/ports.conf #Buscar no ficheiro /etc/apache2/ports.conf mediante o comando *grep* o patrón de texto *'Listen*
root@kaliA:~# nc -vz localhost 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.
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) na IP 192.168.120.100, 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.

b. **Listen 3800 ➜**Todas as interfaces do servidor a escoita no porto TCP 3800 ademais do porto TCP 80):

root@kaliA:~# sed -i 's/^Listen 80/Listen 80\nListen 3800/' /etc/apache2/ports.conf #Pór debaixo da liña Listen 80, outra liña co contido Listen 3800 para indicar que agora o servidor web Apache tamén está a escoita no porto TCP 3800
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz localhost 80 3800 #Mediante o comando nc(netcat) comprobar se os portos 80 e 3800 do servidor web Apache están 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. Os números 80 e 3800 son os portos TCP a escanear.
root@kaliA:~# nc -vz 192.168.120.100 80 3800 #Mediante o comando nc(netcat) comprobar se os portos 80 e 3800 do servidor web Apache están en estado escoita(listen) na IP 192.168.120.100, 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. Os números 80 e 3800 son os portos TCP a escanear.

c. **Listen IP:4300 ➜** IP escoita no porto TCP 4300

root@kaliA:~# echo 'Listen 192.168.120.100:4300' >> /etc/apache2/ports.conf #Engadir a liña *Listen 192.168.120.100:4300* no ficheiro /etc/apache2/ports.conf para indicar que agora o servidor web Apache tamén está a escoita no porto TCP 4300 na IP 192.168.120.100
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz 192.168.120.100 4300 #Mediante o comando nc(netcat) comprobar se o port 4300 do servidor web Apache está en estado escoita(listen) na IP 192.168.120.100, 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 porto solicitado. O número 4300 é o ports TCP a escanear.

d. **Listen IP:8443 http ➜** IP escoita no porto TCP 8443 atendendo o protocolo HTTP

root@kaliA:~# echo 'Listen 192.168.120.100:8443 http' >> /etc/apache2/ports.conf #Engadir a liña *Listen 192.168.120.100:8443 http* no ficheiro /etc/apache2/ports.conf para indicar que agora o servidor web Apache tamén está a escoita no porto TCP 8443 na IP 192.168.120.100 o protocolo http
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz 192.168.120.100 8443 #Mediante o comando nc(netcat) comprobar se o port 8443 do servidor web Apache está en estado escoita(listen) na IP 192.168.120.100, 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 porto solicitado. O número 8443 é o ports TCP a escanear.

13. **Exemplo2. Aloxar Cartafoles.**

Na instalación defínese na directiva DocumentRoot o cartafol onde Apache aloxa as páxinas, sendo este: /var/www/html/, de tal xeito que incorporando ficheiros e cartafoles dentro desa ruta poderase acceder ao contido aloxado nos mesmos.

1. Acceder ao servidor web e crear/copiar varios ficheiros(cartafoles) en /var/www/html/

   root@kaliA:~# cat > /var/www/html/info.php <<EOF #Comezo do ficheiro a creari
   <?php Comezo do código PHP
   phpinfo(); Función phpinfo(), a cal amosa información sobre a configuración de PHP
   ?> Fin do código PHP
   EOF Fin do ficheiro a crear /var/www/html/info.php
   root@kaliA:~# cp -pv /var/www/html/index.html /var/www/html/index2.html #Copiar o ficheiro /var/www/html/index.html en /var/www/html/index2.html en modo verbose (detallado) e preservando permisos e datas.
   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/info.php #Cambiar a só lectura os permisos **ugo** do ficheiro info.php 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)

2. Acceder dende calquer equipo cliente(kaliB) ás seguintes direccións web:

   http://192.168.120.100/index2.html

   http://192.168.120.100/info.php

IMPORTANTE: Como se pode observar as páxinas cargan sen ter que reiniciar o servidor:

root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.

Isto é debido a que o Servidor Web Apache está activo e sempre está a exportar o valor da directiva **DocumentRoot**. Polo tanto como **DocumentRoot** toma o valor **/var/www/html/** namentras o Servidor Web Apache estea activo todo o que aí se garda estará exposto polo servidor.

14. **Exemplo3. Xerar virtualhost: Virtualhost baseados en nome.**
    1. Engadir no directorio /etc/apache2/sites-available/ os seguintes bloques de configuración de virtualhosts. Cada bloque pertence a un arquivo .conf:

Arquivo empresa1.conf

```
#Configuración virtualhost: empresa1
<VirtualHost *:80>
DocumentRoot /var/www/empresa1/
ServerName www.empresa1.com
ServerAlias empresa1.com empresa1.es www.empresa1.es
</VirtualHost>
```

Arquivo empresa2.conf

```
#Configuración virtualhost: empresa2
<VirtualHost *:80>
DocumentRoot /var/www/empresa2/
ServerName www.empresa2.com
ServerAlias empresa2.com empresa2.es www.empresa2.es
</VirtualHost>
```

Explicación bloques configuración virtualhost:

- <VirtualHost *:80> ➜ Inicio etiqueta virtualhost.
- DocumentRoot /var/www/empresa1/ ➜ Definición da ruta onde está aloxada a páxina web no servidor, neste caso: /var/www/empresa1/ mediante a directiva DocumentRoot.
- ServerName www.empresa1.com ➜ Definición do nome DNS que buscará a páxina aloxada na ruta anterior do servidor mediante a directiva ServerName. É o nome que escribes no navegador para visitar a páxina.
- ServerAlias empresa1.com ➜ A directiva ServerAlias permite definir outros nomes DNS para a mesma páxina.
- </VirtualHost> ➜ Fin da etiqueta VirtualHost: fin da definición deste virtualhost para empresa1.

    2. Xerar os directorios /var/www/empresa1 e /var/www/empresa2, os ficheiros index.html dentro deles e establecer permisos para que Apache poida acceder a eses ficheiros index.html.

```
root@kaliA:~# mkdir /var/www/empresa1 /var/www/empresa2 #Crear os directorios /var/www/empresa1 e /var/www/empresa2
root@kaliA:~# echo 'empresa1 contido' > /var/www/empresa1/index.html Crear o ficheiro /var/www/empresa1/index.html co contido: empresa1 contido
root@kaliA:~# echo 'empresa2 contido' > /var/www/empresa2/index.html Crear o ficheiro /var/www/empresa2/index.html co contido: empresa2 contido
root@kaliA:~# chown -R www-data: /var/www/empresa1 /var/www/empresa2 #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan dos directorios /var/www/empresa1 e /var/www/empresa2
```

    3. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite empresa1 Comando que permite habilitar a configuración do VirtualHost empresa1, é dicir, comando que permite habilitar o ficheiro do VirtualHost empresa1 situado no directorio /etc/apache2/sites-available/empresa1.conf engadindo a ligazón correspondente dende /etc/apache2/sites-enabled/empresa1.conf a /etc/apache2/sites-available/empresa1.conf
root@kaliA:~# a2ensite empresa2 Comando que permite habilitar a configuración do VirtualHost empresa2, é dicir, comando que permite habilitar o ficheiro do VirtualHost empresa2 situado no directorio /etc/apache2/sites-available/empresa2.conf engadindo a ligazón correspondente dende /etc/apache2/sites-enabled/empresa2.conf a /etc/apache2/sites-available/empresa2.conf
```

    4. Acceder dende o equipo cliente kaliB ás seguintes direccións web:

http://192.168.120.100/empresa1/index.html
http://192.168.120.100/empresa2/index.html

NOTA: Como se pode observar agora as páxinas non cargan, porque o DocumentRoot non toma o valor /var/www/html senón os valores /var/www/empresa1 e /var/www/empresa2 para empresa1 e empresa2 respectivamente (configurados nos VirtualHost correspondentes en sites-available), e polo tanto, non están activos no servidor na raíz do VirtualHost por defecto (000-default.conf)

5. Actualizar o arquivo /etc/hosts no cliente kaliB:

root@kaliA:~# exit #Saír da consola remota ssh na que estamos a traballar, para voltar á consola local do usuario **kali** na máquina kaliB.
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kaliB:~# echo '192.168.120.100 www.empresa1.com empresa1.com empresa1.es www.empresa1.es' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) os nomes www.empresa1.com, empresa1.com, empresa1.es e www.empresa1.es para que atendan á IP 192.168.120.100
root@kaliB:~# echo '192.168.120.100 www.empresa2.com empresa2.com empresa2.es www.empresa2.es' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) os nomes www.empresa2.com, empresa2.com, empresa2.es e www.empresa2.es para que atendan á IP 192.168.120.100
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kaliB:~$

6. Acceder de novo dende o equipo cliente kaliB ás seguintes direccións web:

http://www.empresa1.com/index.html
http://empresa1.com/index.html
http://empresa1.es/index.html
http://www.empresa1.es/index.html

http://www.empresa2.com/index.html
http://empresa2.com/index.html
http://empresa2.es/index.html
http://www.empresa2.es/index.html

IMPORTANTE: Como se pode observar agora as páxinas non cargan, porque aínda que actualizamos a configuración dos sitios(VirtualHost) do Servidor Web Apache, é necesario recargar o servidor para que se atenda á nova configuración realizada. Así, temos que recargar o servidor:

root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

7. Unha vez recargada a configuración do Servidor Web Apache acceder de novo dende o equipo cliente kaliB ás anteriores direccións web:

root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

http://www.empresa1.com/index.html
http://empresa1.com/index.html
http://empresa1.es/index.html
http://www.empresa1.es/index.html

http://www.empresa2.com/index.html
http://empresa2.com/index.html
http://empresa2.es/index.html
http://www.empresa2.es/index.html

IMPORTANTE: a2ensite + reload ➜ Unha ver activado os sitios(a2ensite) e recargado(reload) o servidor as páxinas(VirtualHost) cargan.

15. **Exemplo4. Control de acceso.**

A. **Control de acceso por HTTP Basic**

Na autenticación HTTP Basic é moi típico utilizar **arquivos .htaccess** nos directorios que queremos controlar o acceso. Os arquivos .htaccess son ficheiros de configuración do propio directorio onde exista.

Para usar arquivos .htaccess, necesítase ter unha configuración no servidor que permita poñer directivas de autenticación nestes arquivos, mediante a directiva AllowOverride, tal como segue: AllowOverride AuthConfig

> **NOTA:** Visitar o seguinte enlace para ver unha explicación, máis polo miúdo, sobre á autenticación http basic: **Autenticación y autorización**

Procedemento:

1. Modificar arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

   <Directory /var/www/html/auth-empresa>
   AllowOverride Authconfig
   </Directory>

2. Crear o contrasinal para o usuario nome_usuario no ficheiro de contrasinais /etc/apache2/web.htpasswd:

   root@kaliA:~# htpasswd -c /etc/apache2/web.htpasswd nome_usuario

3. Crear o directorio /var/www/html/auth-empresa:

   root@kaliA:~# mkdir /var/www/html/auth-empresa

4. Configuralo servidor para o acceso sexa permitido mediante autenticación: usuario/contrasinal empregando un arquivo .htaccess:

   root@kaliA:~# cat /var/www/html/auth-empresa/.htaccess Amosar contido arquivo .htacesss

   ```
   AuthType Basic
   AuthName "Web con Autenticacion Basic"
   AuthBasicProvider file
   AuthUserFile /etc/apache2/web.htpasswd
   ##Require valid-user
   Require user nome_usuario
   ```

5. Xerar dentro do directorio /var/www/html/auth-empresa o ficheiro secret.txt e establecer permisos para que Apache poida acceder a ese ficheiro secret.txt

   root@kaliA:~# echo 'S3c3eT contido' > /var/www/html/auth-empresa/secret.txt Crear o ficheiro /var/www/html/auth-empresa/secret.txt co contido: *S3cr3T contido*
   root@kaliA:~# chown -R www-data: /var/www/html/auth-empresa #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio /var/www/html/auth-empresa
   root@kaliA:~# chmod 400 /var/www/html/auth-empresa/.htaccess #Cambiar a só lectura os permisos **ugo** do ficheiro .htaccess situado en /var/www/html/auth-empresa, é dicir, establecer os permisos r-------- (soamente lectura para o usuario propietario)

6. Actualizar a configuración de Apache para ter en conta os novos cambios:

   root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

7. Actualizar o arquivo /etc/hosts no cliente kaliB:

```
root@kaliA:~# exit #Saír da consola remota ssh na que estamos a traballar, para voltar á consola local do usuario kali na máquina kaliB.
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kaliB:~# echo '192.168.120.100 auth-empresa.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome auth-empresa.local para que atenda á IP 192.168.120.100
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
kali@kaliB:~$
```

8. Acceder de novo dende o equipo cliente kaliB á seguinte dirección web:

http://auth-empresa.local

IMPORTANTE: Como estamos a empregar o sitio por defecto de Apache (DocumenRoot ➜ /var/www/html), aínda que configuramos a autenticación, durante todo o proceso o arquivo secret.txt antes de recargar o servidor estivo visible e accesible. Mellor sería ter configurado isto mediante VirtualHost, tal que non se activaría o sitio ata que executaramos o comando a2ensite correspondente:

a2ensite + reload ➜ Unha ver activado o sitio(a2ensite) e recargado(reload) o servidor a páxina(VirtualHost) carga.

B. Control de acceso por IP (Order, Deny, Allow)

1. Tamén pódese controlar o acceso mediante IP. No seguinte exemplo IP_permiso_concedido define a IP que únicamente ten permiso de acceso. Copiamos este bloque ao arquivo **/etc/apache2/sites-available/controlIP.conf**

```
<VirtualHost *:80>
Alias /cartafol-controlado "/var/www/control/cartafol-controlado/"
<Directory "/var/www/control/cartafol-controlado/">
Order deny,allow
Deny from all
#Allow from IP_permiso_concedido
Allow from 192.168.120.101
</Directory>
DocumentRoot /var/www/control/cartafol-controlado/
ServerName www.empresa.local
ServerAlias empresa.local
</VirtualHost>
```

> Actualmente as directivas: Order, Deny e Allow están en desuso e xa non son necesarias. Son sustituidas pola directiva Require e o contenedor <RequireAll>

2. Crear o seguinte contido:

```
root@kaliA:~# mkdir -p /var/www/control/cartafol-controlado #Crear a estrutura arbórea de directorios ata inclusive o directorio cartafol-compartido
root@kaliA:~# echo 'Contido control' > /var/www/control/cartafol-controlado/control.txt #Crear o ficheiro control.txt no directorio anterior (/var/www/control/cartafol-controlado ➜ DocumentRoot)
```

3. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite controlIP Comando que permite habilitar a configuración do VirtualHost controlIP, é dicir, comando que permite habilitar o ficheiro do VirtualHost controlIP situado no directorio /etc/apache2/sites-available/controlIP.conf engadindo a ligazón correspondente dende /etc/apache2/sites-enabled/controlIP.conf a /etc/apache2/sites-available/controlIP.conf
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

4. Lanzar no navegador da máquina virtual B (KaliB) unha nova lapela coa URL **http://192.168.120.100/control/cartafol-controlado/control.txt** Que acontece? Por que?
   Pois obtemos erro porque agora non se espera na IP ese cartafol senón un nome DNS por medio do VirtualHost xerado. Así deberiamos crear en /etc/hosts unha entrada como a seguinte:

   ```
   192.168.120.100 empresa.local www.empresa.local
   ```

   de tal xeito que se visitaramos http://empresa.local veriamos o esperado.

5. E se visitamos a URL **http://kaliA/control/cartafol-controlado/control.txt** ? Pois máis do mesmo, porque o nome DNS non se corresponde e polo tanto visitaría o DocumentRoot de 000-default, e como /var/www/html/control/cartafol-controlado non existe, obteriamos erro.

6. E se visitamos a URL **http://empresa.local** ?
   Pois agora si que visitariamos o esperado.

7. E se visitamos a URL **http://empresa.local/cartafol-controlado** ?
   Pois seguiriamos vendo o esperado, xa que existe un Alias definido no VirtualHost de xeito que somos redireccionados a /var/www/control/cartafol-controlado, visualizando o contido esperado.

C. **Control de acceso por IP (Require e <RequireAll>)**

1. Imos facer de novo a opción B (controlar o acceso mediante IP) pero agora empregando a directiva aconsellada por Apache: **Require** (<RequireAll>). Así, modificamos o anterior bloque VirtualHost do arquivo **/etc/apache2/sites-available/controlIP.conf** tal como segue:

```
<VirtualHost *:80>
Alias /cartafol-controlado "/var/www/control/cartafol-controlado/"
<Directory "/var/www/control/cartafol-controlado/">
#Order deny,allow
#Deny from all
#Allow from 192.168.120.101
#Require ip IP_permiso_concedido
Require ip 192.168.120.101
</Directory>
DocumentRoot /var/www/control/cartafol-controlado/
ServerName www.empresa.local
ServerAlias empresa.local
</VirtualHost>
```

2. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite controlIP Comando que permite habilitar a configuración do VirtualHost controlIP, é dicir, comando que permite habilitar o ficheiro do VirtualHost controlIP situado no directorio /etc/apache2/sites-available/controlIP.conf engadindo a ligazón correspondente dende /etc/apache2/sites-enabled/controlIP.conf a /etc/apache2/sites-available/controlIP.conf
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

3. Lanzar no navegador da máquina virtual B (KaliB) unha nova lapela coa URL **http://192.168.120.100/control/cartafol-controlado/control.txt** Que acontece? Por que?
   Pois obtemos erro porque agora non se espera na IP ese cartafol senón un nome DNS por medio do VirtualHost xerado. Así deberiamos ter en /etc/hosts unha entrada como a seguinte:

   ```
   192.168.120.100 empresa.local www.empresa.local
   ```

   de tal xeito que se visitaramos http://empresa.local veriamos o esperado.

4. E se visitamos a URL **http://kaliA/control/cartafol-controlado/control.txt** ? Pois máis do mesmo, porque o nome DNS non se corresponde e polo tanto visitaría o DocumentRoot de 000-default, e como /var/www/html/control/cartafol-controlado non existe, obteriamos erro.

5. E se visitamos a URL **http://empresa.local** ?
   Pois agora si que visitariamos o esperado.

6. E se visitamos a URL **http://empresa.local/cartafol-controlado** ?
   Pois seguiriamos vendo o esperado, xa que existe un Alias definido no VirtualHost de xeito que somos redireccionados a /var/www/control/cartafol-controlado, visualizando o contido esperado.

16. **Exemplo5. Prioridade seccións/directivas:**

    Á hora de configurar o servidor web Apache temos que ter en conta que é o que acontece cando unha mesma directiva pertence a distintas seccións. Cal é a directiva que se atende? Cal é a prioridade? Imos revisar a prioridade empregado a directiva Options co argumento Indexes:

    - Options +Indexes ➜ no caso de non existir un index.html permite ver o contido do cartafol visitado.
    - Options -Indexes ➜ no caso de non existir un index.html non permite ver o contido do cartafol visitado amosando o erro 403(Forbidden).

    a. Crear o seguinte contido:

        root@kaliA:~# mkdir /var/www/html/prioridade #Crear o directorio prioridade no DocumentRoot(/var/www/html) do sitio por defecto que configura Apache (000-default.conf)
        root@kaliA:~# echo 'Contido f1.txt' > /var/www/html/prioridade/f1.txt #Crear o ficheiro f1.txt no directorio anterior (/var/www/html/prioridade ➜ DocumentRoot)

    b. Visitar http://localhost/prioridade

    c. Modificar arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

        Sección Directory

        <Directory /var/www/html/prioridade>
        Options -Indexes
        </Directory >

        Recargar a configuración:

        root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

    d. Visitar de novo http://localhost/prioridade
    Agora non se pode visualizar a páxina amosándose o erro 403 Forbidden

    e. Modificar de novo o arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

        Sección Location

        <Location /prioridade>
        Options +Indexes
        </Location >

    f. Visitar de novo http://localhost/prioridade
    Agora si se pode visualizar a páxina

Ricardo Feijoo Costa

Cifrado asimétrico: Certificado Apache

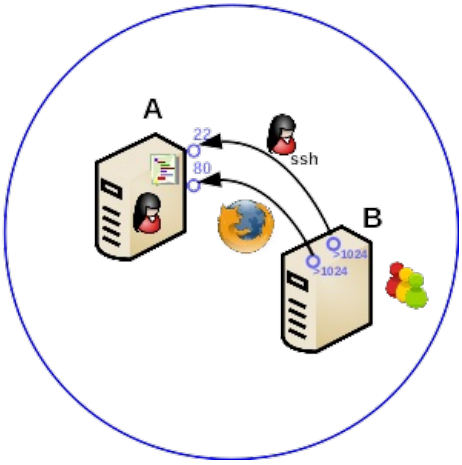
ESCENARIO

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

Máquina virtual A:
Rede Interna
Servidor SSH: openssh-server
Servidor Web: Apache (apache2)
ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
Cliente Web: Navegador (firefox)
ISO: Kali Live amd64
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:

- Cliente ssh GNU/Linux: **comando ssh (paquete openssh-client)**
- Documentación oficial sobre o Servidor web **Apache (v2.4)**

Práctica Cifrado asimétrico: Certificado Apache

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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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 á máquina virtual A dende a máquina virtual B. A partir de 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. Lanzar no navegador da máquina virtual B (Kali) unha nova lapela coa URL **https://192.168.120.100** Que acontece? Por que?
E se visitamos a URL **https://kaliA** Que acontece? Por que?

13. Activar configuración https (módulo SSL, porto TCP 443) en Apache:

Revisar o contido dos directorios:
- /etc/apache2/sites-available
- /etc/apache2/sites-enabled

root@kaliA:~# nc -vz 192.168.120.100 443 #Mediante o comando nc(netcat) comprobar se o porto 443 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 443 é o porto TCP a escanear.
root@kaliA:~# a2ensite default-ssl #Habilitar o VirtualHost default-ssl, que configura o acceso a través de https (porto TCP 443)
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do servidor web Apache.

Revisar o contido dos directorios:
- /etc/apache2/sites-available
- /etc/apache2/sites-enabled

root@kaliA:~# nc -vz 192.168.120.100 443 #Mediante o comando nc(netcat) comprobar se o porto 443 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 443 é o porto TCP a escanear.

14. Lanzar de novo no navegador da máquina virtual B (Kali) unha nova lapela coa URL **https://192.168.120.100** Que acontece? Por que?
E se visitamos a URL **https://kaliA** Que acontece? Por que?

15. Activar certificado https (módulo SSL, porto TCP 443) en Apache:

```
root@kaliA:~# nc -vz 192.168.120.100 443 #Mediante o comando nc(netcat) comprobar se o porto 443 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 443 é o porto TCP a escanear.
root@kaliA:~# a2enmod ssl #Habilitar o módulo ssl que permite activar a configuración do VirtualHost default-ssl, que configura o acceso a través de https (porto TCP 443)
root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar a configuración do servidor web Apache.
root@kaliA:~# nc -vz 192.168.120.100 443 #Mediante o comando nc(netcat) comprobar se o porto 443 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 443 é o porto TCP a escanear.
```

16. Lanzar de novo no navegador da máquina virtual B (Kali) unha nova lapela coa URL **https://192.168.120.100** Que acontece? Por que?
    E se visitamos a URL **https://kaliA** Que acontece? Por que?

17. É a conexión segura? A transmisión da información realízase mediante cifrado (RSA, DSA...) (MD5, SHA1, SHA-256...)? O navegador empregado confía no certificado configurado no servidor Apache? Ese certificado está asinado por unha entidade certificadora? Podemos ver información sobre o certificado de Apache a través do navegador(Si/Non)? Se é posible como se pode revisar esa información?

Ricardo Feijoo Costa

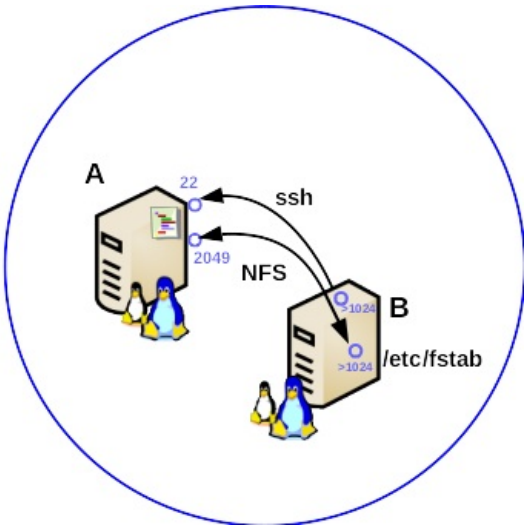
Servizo NFS (TCP wrappers + /etc/exports)

ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0/24
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede Interna e NAT
Servidor SSH: openssh-server
Servidor NFS: nfs-kernel-server
ISO: Kali amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client(ssh)
Cliente NFS: nfs-common
ISO: Kali amd64
IP/MS: 192.168.120.101/24



NOTAS:

- O sistema de arquivos en rede **NFS** posúe 3 versións NFSv2, NFSv3 e NFSv4 e é comunmente empregado para compartir entre sistemas GNU/Linux. NFS permite:
  a. Compartir cartafoles a equipos ou a redes (Todas as versións).
  b. Control de acceso: ACL mediante TCP wrappers (/etc/hosts.allow, /etc/hosts.deny) (Todas as versións)
  c. Compartir mediante permisos de lectura/escritura (rw) ou soamente lectura (ro) (Todas as versións).
  d. Compartir mediante autenticación usuarios/grupos e cifrado (Soamente versión 4 mediante Kerberos).

  - **As versión NFSv2 e NFSv3:**
    - Non son compatibles con Kerberos.
    - Empregan portmap(rpcbind) para poder compartir, o cal asigna as peticións de chamadas de procedementos remotos (RPC) aos servizos correctos.

      Empregar o comando **rpcinfo -p** para revisar o estado dos servizos NFS basados en RPC.
    - Poden empregar os protocolos TCP ou UDP:
      - NFS: 2049
      - portmap(rpcbind): 111 e outros
  - **A versión NFSv4:**
    - Emprega Kerberos.
    - Non emprega portmap(rpcbind).
    - Soamente emprega o protocolo TCP no porto 2049.

- Para conectar con este servizo imos empregar:
  a. Control de acceso mediante TCP wrappers: ficheiros **/etc/hosts.allow** e **/etc/hosts.deny**.

  TCP wrappers
  Sóese controlar o acceso ao servizo NFS mediante **listas de control de acceso (ACL)** definidas nos arquivos: **/etc/hosts.allow** e **/etc/hosts.deny**:

  - No arquivo **/etc/hosts.allow** defínense as **ACL** para permitir o acceso e no **/etc/hosts.deny** para denegar.
  - Primeiro lesen secuencialmente, de arriba-abaixo, as **ACL** do arquivo **/etc/hosts.allow**, de tal xeito que se unha ACL coincide non se segue lendo no propio arquivo e tampouco lese o arquivo **/etc/hosts.deny**.
  - Hai que ter en conta:
    - Que se non se atopa ningunha regra coincidente permítese o acceso ao servizo.
    - Calquera cambio nos arquivos lesen inmediatamente sen ter que reiniciar o servizo.

  b. Para **compartir sistemas de ficheiros** o ficheiro **/etc/exports** e o comando **exportfs**.

  Prioridade
  Primeiro lense os TCP wrappers (/etc/hosts.allow e /etc/hosts.deny) e se estes permiten o acceso será lido o ficheiro /etc/exports.

  c. Para **verificar a compartición dos sistemas de ficheiros** o comando **showmount**.
  d. Para montaxe de sistemas de ficheiros de forma manual **comandos** na consola como: **mount** e **mount.nfs**.
  e. Para montaxe de sistemas de ficheiros de forma automática e permanentes o ficheiro **/etc/fstab** (man fstab && man nfs).

- Cliente NFS: Paquete nfs-common (# apt update && apt -y install nfs-common)
- Servidor NFS: Paquete nfs-kernel-server (# apt update && apt -y install nfs-kernel-server)
- Ficheiros de configuración do servidor:
  - TCP wrappers:
    - **/etc/hosts.allow** (man hosts_allow || man 5 hosts_access)
    - **/etc/hosts.deny** (man hosts_deny || man 5 hosts_access)
  - **/etc/exports** (man exports)
  - **/etc/default/nfs-kernel-server**
- Cliente SSH: Comando ssh. Paquete openssh-client (# apt update && apt -y install openssh-client).
- Servidor SSH: Paquete openssh-server (# apt update && apt -y install openssh-server).

NOTAS: **Permisos de compartición.**

- **ro:** Acceso read only (so lectura).
- **rw:** Acceso read write (lectura/escritura).
- **no_subtree_check:** Non comprobar que se accede á zona compartida polo servidor, xa que ralentiza moito o acceso. Para evitar ter que activar esta opción o mellor e crear zonas compartidas en cartafoles independentes no directorio raíz, de tal xeito que asi impedimos xa o acceso a cartafoles fora do recurso compartido -facendo unha similitude de funcionamento sen ter que activar esta opción-. Esta opción está activada por defecto no servidor NFS e non é necesario indicala.
- **async:** Permite ao servidor escribir os datos no disco a intervalos irregulares. Funciona mellor para accesos ro
- **sync:** Tódalas escrituras no disco do servidor fanse antes de que a petición de escritura do cliente remate, evitando así perda de datos en caso de desconexións. Funciona mellor para accesos rw pero pode disminuir o rendemento.

IMPORTANTE

- NFSv2 e NFSv3: Os privilexios de montaxe son otorgados ao host cliente non ao usuario. é dicir, **calquera usuario do sistema cliente ten acceso aos sistemas de ficheiros compartidos tal cual foron exportados**. Polo tanto, revisar cales son os sistemas de ficheiros que deben posuír permisos exportados **rw** nos hosts clientes.

Servizo NFS (NFSv2 e NFSv3)

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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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** ^{SSH} **➜ 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. **Instalar Servidor NFS**:

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)

root@kaliA:~# apt search nfs-kernel-server #Buscar calquera paquete que coincida co patrón de búsqueda nfs-kernel-server

root@kaliA:~# apt -y install nfs-kernel-server #Instalar o paquete nfs-kernel-server, é dicir, instalar o servidor HTTP nfs-kernel-server. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

9. **Activar Servidor NFS**:

root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.
root@kaliA:~# /etc/init.d/nfs-kernel-server start #Iniciar o servidor NFS.
root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.
root@kaliA:~# nc -vz 192.168.120.100 2049 #Mediante o comando nc(netcat) comprobar se o porto 2049 do servidor NFS 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 2049 é o porto TCP a escanear.

10. **Crear estrutura a exportar**:

root@kaliA:~# dd if=/dev/zero of=file1.raw bs=1MiB count=100 #Crear un ficheiro file1.raw que contén todos ceros no directorio actual cun tamaño de 100MiB.
root@kaliA:~# losetup -f --show file1.raw #Enlazar a file1.raw o primeiro dispositivo loop libre (-f), e amosando cal é (--show).
root@kaliA:~# losetup -a #Amosar tódolos dispositivos loop enlazados.
root@kaliA:~# parted --script /dev/loop1 mklabel msdos #Crear a etiqueta de disco ao dispositivo /dev/loop1 sen ter que acceder ao prompt de parted
root@kaliA:~# parted --script /dev/loop1 mkpart primary 0 50% #Crear unha partición primaria co primeiro 50% do dispositivo /dev/loop1 sen ter que acceder ao prompt de parted
root@kaliA:~# parted --script /dev/loop1 mkpart primary 50% 100% #Crear unha partición primaria co último 50% do dispositivo /dev/loop1 sen ter que acceder ao prompt de parted
root@kaliA:~# ls -lah /dev/loop1* #Listar o dispositivo /dev/loop1 e as súas particións (xeradas anteriormente): /dev/loop1p1, /dev/loop1p2
root@kaliA:~# mkfs.ext4 -L 'parte1loop1' /dev/loop1p1 #Formatear en ext4 a partición /dev/loop1p1 etiquetada como parte1loop1.
root@kaliA:~# mkfs.ext4 -L 'parte2loop1' /dev/loop1p2 #Formatear en ext4 a partición /dev/loop1p2 etiquetada como parte2loop1.
root@kaliA:~# mkdir -p /media/loop1/loop1p1 /media/loop1/loop1p2 #Crear cartafoles /media/loop1/loop1p1 e /media/loop1/loop1p2
root@kaliA:~# mount /dev/loop1p1 /media/loop1/loop1p1 #Montar /dev/loop1p1 en /media/loop1/loop1p1
root@kaliA:~# mount /dev/loop1p2 /media/loop1/loop1p2 #Montar /dev/loop1p2 en /media/loop1/loop1p2
root@kaliA:~# mount | grep loop #Amosar dispositivos montados que concordan co patrón loop
root@kaliA:~# cp -pv /etc/passwd /media/loop1/loop1p1 #Copiar o ficheiro /etc/passwd en /media/loop1/loop1p1
root@kaliA:~# echo 'Ficheiro exportado' > /media/loop1/loop1p1/fexportado.txt #Crear o ficheiro fexportado.txt dentro do directorio /media/loop1/loop1p1 e co contido: *Ficheiro exportado*.
root@kaliA:~# ls -ld /media/loop1/loop1p1 #Listar soamente os permisos do cartafol /media/loop1/loop1p1, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
root@kaliA:~# ls -l /media/loop1/loop1p1/fexportado.txt #Listar de forma extendida (tipo atopado, permisos, propietarios...) o ficheiro /media/loop1/loop1p1/fexportado.txt

11. **Exemplo1. Compartición NFS - Permisos ro (so lectura)**

i. Arquivo **/etc/exports** tipo:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
##Compartir por nome/dominios. Pódense empregar comodíns: *, ?
##/srv *.edu.xunta.es(ro,sync,no_subtree_check)
##Compartir a unha IP
##/srv 192.168.120.101(ro,sync,no_subtree_check)
##Compartir a varias IP
##/srv 192.168.120.101(ro,sync,no_subtree_check)
192.168.120.102(rw,sync,no_subtree_check)
#Compartir a unha rede
/media/loop1/loop1p1 192.168.120.0/24(ro,sync)
```

ii. Aplicar cambios do ficheiro /etc/exports:

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportandos mediante NFS. De momento, como non actualizamos os cambios de configuración, non temos ningún exportado.
root@kaliA:~# exportfs -av #Ler o ficheiro /etc/exports e exportar o indicado na súa configuración. A opción -a permite exportar e a opción -v é o modo verbose(detallado), é dicir, amosar máis información na saída de execución do comando.

> Se a opción **-a** emprégase coa opción **-u** elimina a exportación de todos os sistemas de ficheiros que estaban exportados.
> # exportfs -auv

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportandos mediante NFS. Agora si, unha vez recargada a configuración, podemos ver os sistemas de ficheiros exportados.

iii. **Cliente: Conexión ao servidor NFS**:

A. **EN TEMPO REAL**

- **mount -t nfs**

  mount -t nfs IP_Servidor:cartafol_compartido ruta_montaxe_local

  root@kaliB:~# mount -t nfs 192.168.120.100:/media/loop1/loop1p1 /mnt/

- **mount.nfs**

  mount.nfs IP_Servidor:cartafol_compartido ruta_montaxe_local

  root@kaliB:~# mount.nfs 192.168.120.100:/media/loop1/loop1p1 /mnt/

B. **DE FORMA PERMANENTE: /etc/fstab**

i. Engadir no ficheiro **/etc/fstab** a seguinte liña:

192.168.120.100:/media/loop1/loop1p1 /mnt nfs soft,intr,rsize=32768,wsize=32768,ro 0 0

Opcións montaxe cliente:

- **hard** ou **soft:** Qué facer no caso de desconexión co servidor NFS. Con hard hai que esperar a que o servidor volva a estar activo e con soft infórmase do erro.
- **intr:** Permite que se interrumpan peticións NFS se se produce desconexión co servidor.
- **rsize=32768, wsize=32768:** Aceleran a comunicación NFS para lecturas *rsize* e escrituras *wsize*. Configuran un tamaño de bloques de datos en bytes para transferir de cada vez.
- **ro:** Read only (so lectura).

ii. Remontar sen ter que reiniciar tódolos puntos de montaxe definidos no **/etc/fstab** :

root@kaliB:~# mount -a

IMPORTANTE (NFSv2 e NFSv3)
Podemos comprobar que aínda que estamos a traballar co usuario **root** no cliente e nos seus permisos **ugo** no directorio **/mnt** posúe permisos de escritura, como o sistema de ficheiros exportados está compartido con permisos de soamente lectura, o usuaro **root** non pode escribir nese directorio.

```
root@kaliB:~# whoami
root
root@kaliB:~# ls -ld /mnt
drwxr-xr-x 3 root root ...
root@kaliB:~# echo 'Grabando...' > /mnt/fgrabando.txt
-bash: /mnt/fgrabando.txt: Read-only file system
```

iii. Desmontar /mnt (sistema de ficheiros compartido):

root@kaliB:~# umount /mnt

12. **Exemplo2. Compartición NFS - TCP wrappers (/etc/hosts.allow, /etc/hosts.deny)**

i. Comentar todas as liñas do arquivo **/etc/hosts.allow**:

root@kaliA:~# sed -i 's/^/#/' /etc/hosts.allow

ii. Engadir ao arquivo **/etc/hosts.deny**:

```
ALL: 192.168.120.101
```

root@kaliA:~# echo 'ALL: 192.168.120.101' >> /etc/hosts.deny

Prioridade
Primeiro lense os TCP wrappers (/etc/hosts.allow e /etc/hosts.deny) e se estes permiten o acceso será lido o ficheiro /etc/exports. Neste caso non temos nada configurado en /etc/hosts.allow co cal segue a lectura a /etc/hosts.deny, e aquí si que temos configurado que o host 192.168.120.101 ten denegado o acceso, polo tanto cando se cumpra esta condición o arquivo /etc/exports non se lerá.

iii. Arquivo **/etc/exports** tipo:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
##Compartir por nome/dominios. Pódense empregar comodíns: *, ?
##/srv *.edu.xunta.es(ro,sync,no_subtree_check)
##Compartir a unha IP
##/srv 192.168.120.101(ro,sync,no_subtree_check)
##Compartir a varias IP
##/srv 192.168.120.101(ro,sync,no_subtree_check)
192.168.120.102(rw,sync,no_subtree_check)
#Compartir a unha rede
/media/loop1/loop1p1 192.168.120.0/24(ro,sync)
```

iv. Aplicar cambios do ficheiro /etc/exports:

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportandos mediante NFS. De momento, como non actualizamos os cambios de configuración, non temos ningún exportado.
root@kaliA:~# exportfs -av #Ler o ficheiro /etc/exports e exportar o indicado na súa configuración. A opción -a permite exportar e a opción -v é o modo verbose(detallado), é dicir, amosar máis información na saída de execución do comando.

Se a opción **-a** emprégase coa opción **-u** elimina a exportación de todos os sistemas de ficheiros que estaban exportados.
exportfs -auv

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportandos mediante NFS. Agora si, unha vez recargada a configuración, podemos ver os sistemas de ficheiros exportados.

v. **Cliente: Conexión ao servidor NFS**:

A. **EN TEMPO REAL**

- **mount -t nfs**

  mount -t nfs IP_Servidor:cartafol_compartido ruta_montaxe_local

  root@kaliB:~# mount -t nfs 192.168.120.100:/media/loop1/loop1p1 /mnt/

- **mount.nfs**

  mount.nfs IP_Servidor:cartafol_compartido ruta_montaxe_local

  root@kaliB:~# mount.nfs 192.168.120.100:/media/loop1/loop1p1 /mnt/

B. **DE FORMA PERMANENTE: /etc/fstab**

i. Engadir no ficheiro **/etc/fstab** a seguinte liña:

192.168.120.100:/media/loop1/loop1p1 /mnt nfs soft,intr,rsize=32768,wsize=32768,ro 0 0

Opcións montaxe cliente:

- **hard** ou **soft:** Qué facer no caso de desconexión co servidor NFS. Con hard hai que esperar a que o servidor volva a estar activo e con soft infórmase do erro.
- **intr:** Permite que se interrumpan peticións NFS se se produce desconexión co servidor.
- **rsize=32768, wsize=32768:** Aceleran a comunicación NFS para lecturas *rsize* e escrituras *wsize*. Configuran un tamaño de bloques de datos en bytes para transferir de cada vez.
- **ro:** Read only (so lectura).

ii. Remontar sen ter que reiniciar tódolos puntos de montaxe definidos no **/etc/fstab** :

root@kaliB:~# mount -a

IMPORTANTE:
Non ten lugar a compartición por rede mediante NFS, xa que se cumpre a condición configurada en /etc/hosts.deny:
ALL: 192.168.120.101
Polo que obtemos un erro ao intentar realizar a montaxe:
`mount.nfs: access denied by server while mounting 192.168.120.100:/media/loop/loop1p1`

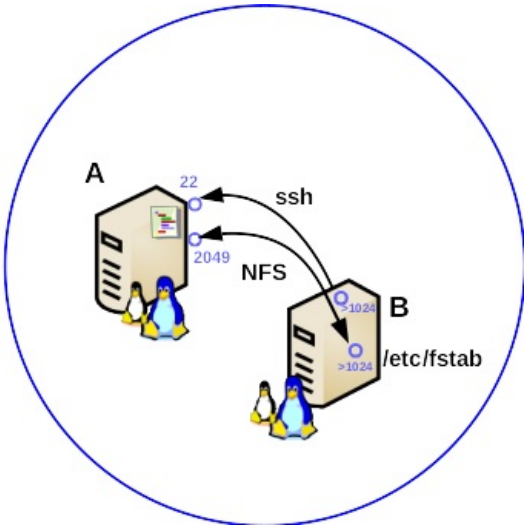
Servizo NFSv4 (~~TCP wrappers~~ + /etc/exports)

ESCENARIO

Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0/24
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede Interna e NAT
Servidor SSH: openssh-server
Servidor NFS: nfs-kernel-server
ISO: Kali amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client(ssh)
Cliente NFS: nfs-common
ISO: Kali amd64
IP/MS: 192.168.120.101/24



NOTAS:

- O sistema de arquivos en rede **NFS** posúe 3 versións NFSv2, NFSv3 e NFSv4 e é comunmente empregado para compartir entre sistemas GNU/Linux. NFS permite:
  a. Compartir cartafoles a equipos ou a redes (Todas as versións).
  b. Control de acceso: ACL mediante TCP wrappers (/etc/hosts.allow, /etc/hosts.deny) (Todas as versións)
  c. Compartir mediante permisos de lectura/escritura (rw) ou soamente lectura (ro) (Todas as versións).
  d. Compartir mediante autenticación usuarios/grupos e cifrado (Soamente versión 4 mediante Kerberos).

    - **As versión NFSv2 e NFSv3:**
      - Non son compatibles con Kerberos.
      - Empregan portmap(rpcbind) para poder compartir, o cal asigna as peticións de chamadas de procedementos remotos (RPC) aos servizos correctos.

        Empregar o comando **rpcinfo -p** para revisar o estado dos servizos NFS basados en RPC.
      - Poden empregar os protocolos TCP ou UDP:
        - NFS: 2049
        - portmap(rpcbind): 111 e outros
    - **A versión NFSv4:**
      - Pode empregar Kerberos.
      - Non emprega portmap(rpcbind).
      - Soamente emprega o protocolo TCP no porto 2049.

- Para conectar con este servizo imos empregar:
  a. ~~Control de acceso mediante TCP wrappers: ficheiros **/etc/hosts.allow** e **/etc/hosts.deny**~~.

    TCP wrappers
    Sóese controlar o acceso ao servizo NFS mediante **listas de control de acceso (ACL)** definidas nos arquivos: **/etc/hosts.allow** e **/etc/hosts.deny**:

    - No arquivo **/etc/hosts.allow** defínense as **ACL** para permitir o acceso e no **/etc/hosts.deny** para denegar.
    - Primeiro lesen secuencialmente, de arriba-abaixo, as **ACL** do arquivo **/etc/hosts.allow**, de tal xeito que se unha ACL coincide non se segue lendo no propio arquivo e tampouco lese o arquivo **/etc/hosts.deny**.
    - Hai que ter en conta:
      - Que se non se atopa ningunha regra coincidente permítese o acceso ao servizo.
      - Calquera cambio nos arquivos lesen inmediatamente sen ter que reiniciar o servizo.

  b. Para **compartir sistemas de ficheiros** o ficheiro **/etc/exports** e o comando **exportfs**.

    Prioridade
    Primeiro lense os TCP wrappers (/etc/hosts.allow e /etc/hosts.deny) e se estes permiten o acceso será lido o ficheiro /etc/exports.

  c. Para **verificar a compartición dos sistemas de ficheiros** o comando **showmount**.
  d. Para montaxe de sistemas de ficheiros de forma manual **comandos** na consola como: **mount -t nfs4** e **mount.nfs4**.
  e. Para montaxe de sistemas de ficheiros de forma automática e permanentes o ficheiro **/etc/fstab** (man fstab && man nfs).

- Cliente NFS: Paquete nfs-common (# apt update && apt -y install nfs-common)

```
$ cat /proc/mounts #Podemos revisar permisos de compartición NFS coa execución deste comando.
```

- Ficheiro de configuración do cliente: **/etc/default/nfs-common (man rpc.idmapd)**

```
$ dpkg -L nfs-common #Listar os ficheiros instalados no sistema para o paquete nfs-common
```

- Servidor NFS: Paquete nfs-kernel-server (# apt update && apt -y install nfs-kernel-server)

- Ficheiros de configuración do servidor:
  - ~~TCP wrappers:~~
    - ~~**/etc/hosts.allow** (man hosts_allow || man 5 hosts_access)~~
    - ~~**/etc/hosts.deny** (man hosts_deny || man 5 hosts_access)~~
  - **/etc/exports** (man exports)
  - **/etc/default/nfs-kernel-server (man rpc.mountd)**

```
$ dpkg -L nfs-kernel-server #Listar os ficheiros instalados no sistema para o paquete nfs-kernel-server
```

- Cliente SSH: Comando ssh. Paquete openssh-client (# apt update && apt -y install openssh-client).

- Servidor SSH: Paquete openssh-server (# apt update && apt -y install openssh-server).

NOTAS: **Permisos de compartición.**

- **ro:** Acceso read only (so lectura).
- **rw:** Acceso read write (lectura/escritura).
- **no_subtree_check:** Non comprobar que se accede á zona compartida polo servidor, xa que ralentiza moito o acceso. Para evitar ter que activar esta opción o mellor e crear zonas compartidas en cartafoles independentes no directorio raíz, de tal xeito que asi impedimos xa o acceso a cartafoles fora do recurso compartido -facendo unha similitude de funcionamento sen ter que activar esta opción-. Esta opción está activada por defecto no servidor NFS e non é necesario indicala.
- **async:** Permite ao servidor escribios datos no disco a intervalos irregulares. Funciona mellor para accesos ro
- **sync:** Tódalas escrituras no disco do servidor fanse antes de que a petición de escritura do cliente remate, evitando así perda de datos en caso de desconexións. Funciona mellor para accesos rw pero pode disminuir o rendemento.
- **crossmnt** en pseudo sistemas de ficheiros raiz (--bind) equivale a opción **nohide** no directorio fillo do pseudo sistemas de ficheiros raíz. As 2 opcións son excluintes. Empregar soamente unha.
- **insecure:** permite aos clientes conectar a portos maiores de 1024. Esta opción non é insegura doutro xeito.
- **fsid=0** ou **fsid=root:** indica a posibilidade de montar aqueles sistemas de ficheiros que non posúen UUID. En NFSv4 o pseudo sistema de ficheiros raíz, do cal colgan todos os sistemas de ficheiros a exportar ten que posuír esta opción para poder ser compartido.
- **_netdev** Montar sempre e cando a rede esté activa.
- **auto** Montar no arranque do sistema. Permitir montar coa opción -a

IMPORTANTE

- NFSv2 e NFSv3: Os privilexios de montaxe son otorgados ao host cliente non ao usuario. é dicir, **calquera usuario do sistema cliente ten acceso aos sistemas de ficheiros compartidos tal cual foron exportados**. Polo tanto, revisar cales son os sistemas de ficheiros que deben posuír permisos exportados **rw** nos hosts clientes.

- NFSv4:
  - A compartición non se fai directamente sobre o cartafol a compartir. Debe existir un directorio raíz onde compartir (pseudo filesystem) e todo o que se quere compartir debe ser fillo deste directorio raíz. Similar á directiva DocumentRoot do Servidor Web Apache, é dicir, agora o cliente non sabe a ruta física exacta do compartido, senón a dirección de compartición.
  - Os permisos NFS non son independentes dos permisos do sistema GNU/Linux: **UGO**, **ACLs**,

prevalecendo os máis rectrictivos. Así, por exemplo, aínda que un cartafol se esté compartindo por NFSv4 con permisos **rw**, se no sistema posee permisos UGO **ro**, prevalecen neste caso os permisos **ro**. É máis, aínda que un cartafol esté compartido con permisos **rw** se un usuario do sistema soamente ten permisos de **ro** nese cartafol, aínda que acceda por NFSv4 ese usuario posúe os permisos **ro**.

Servizo NFSv4

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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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** ^{SSH} **➜ 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. **Instalar Servidor NFS**:

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)

root@kaliA:~# apt search nfs-kernel-server #Buscar calquera paquete que coincida co patrón de búsqueda nfs-kernel-server

root@kaliA:~# apt -y install nfs-kernel-server #Instalar o paquete nfs-kernel-server, é dicir, instalar o servidor NFS nfs-kernel-server. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

9. **Configurar soamente a activación da versión 4 no Servidor NFS**:

root@kaliA:~# cat /proc/fs/nfsd/versions #Revisar no kernel que versións de NFS están permitidas, activadas e desactivadas

-2 +3 +4 +4.1 +4.2 indica que o kernel permite as versións 2, 3, 4, 4.1 e 4.2 estando a versión 2 deshabilitada.

root@kaliA:~# sed -i 's/RPCMOUNTDOPTS="--manage-gids"/RPCMOUNTDOPTS="--manage-gids -V 4 -N 3 -N 2"/' /etc/default/nfs-kernel-server #Modificar o arquivo de configuración /etc/default/nfs-kernel-server para activar soamente a versión NFSv4, é dicir, desactivar as versións 2, 3 e activar a versión 4 do servidor NFS

root@kaliA:~# sed -i -e 's/# vers2=y/vers2=n/' -e 's/# vers3=y/vers3=n/' /etc/nfs.conf #Modificar o arquivo de configuración /etc/nfs.conf para activar soamente a versión NFSv4, é dicir, desactivar as versións 2, 3 e activar a versión 4 do servidor NFS

grep 'vers[0-9]=' /etc/nfs.conf indica que versións están habilitadas(y) e deshabilitadas(n).

/etc/default/nfs-kernel-server

- NEED_SVCGSSD=no ➜ Desactivada seguridade NFSv4.
- RPCMOUNTDOPTS= ➜ Configurar as opcións de montaxe (man rpc.mountd)

10. **Activar Servidor NFS**:

root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.

root@kaliA:~# /etc/init.d/nfs-kernel-server reload #Recargar a configuración do servidor NFS.

root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.

root@kaliA:~# nc -vz 192.168.120.100 2049 #Mediante o comando nc(netcat) comprobar se o porto 2049 do servidor NFS 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 2049 é o porto TCP a escanear.

11. **Crear estrutura a exportar**:

Os sistemas de ficheiros exportados en NTFSv4 soamente existen nun pseudo sistema de ficheiros e logo son remontados mediante a opción bind. A opción --bind permite voltar a montar unha xerarquía de arquivos noutra ruta, tendo agora posibilidade de acceder aos mesmos contidos en 2 lugares diferentes.

root@kaliA:~# dd if=/dev/zero of=file1.raw bs=1MiB count=100 #Crear un ficheiro file1.raw que contén todos ceros no directorio actual cun tamaño de 100MiB.

root@kaliA:~# losetup -f --show file1.raw #Enlazar a file1.raw o primeiro dispositivo loop libre (-f), e amosando cal é (--show).

root@kaliA:~# losetup -a #Amosar tódolos dispositivos loop enlazados.

root@kaliA:~# parted --script /dev/loop1 mklabel msdos #Crear a etiqueta de disco ao dispositivo /dev/loop1 sen ter que acceder ao prompt de parted

root@kaliA:~# parted --script /dev/loop1 mkpart primary 0 50% #Crear unha partición primaria co primeiro 50% do dispositivo /dev/loop1 sen ter que acceder ao prompt de parted

root@kaliA:~# parted --script /dev/loop1 mkpart primary 50% 100% #Crear unha partición primaria co último 50% do dispositivo /dev/loop1 sen ter que acceder ao prompt de parted

root@kaliA:~# ls -lah /dev/loop1* #Listar o dispositivo /dev/loop1 e as súas particións (xeradas anteriormente): /dev/loop1p1, /dev/loop1p2

root@kaliA:~# mkfs.ext4 -L 'parte1loop1' /dev/loop1p1 #Formatear en ext4 a partición /dev/loop1p1 etiquetada como parte1loop1.

root@kaliA:~# mkfs.ext4 -L 'parte2loop1' /dev/loop1p2 #Formatear en ext4 a partición /dev/loop1p2 etiquetada como parte2loop1.

root@kaliA:~# mkdir -p /media/loop1/loop1p1 /media/loop1/loop1p2 #Crear cartafoles /media/loop1/loop1p1 e /media/loop1/loop1p2

root@kaliA:~# export COMPARTIR_1='/media/loop1/loop1p1' #Declara unha variable de nome **COMPARTIR1** co valor **/media/loop1/loop1p1** e ademais exporta a variable para que poida ser recoñecida na contorna actual da shell.
root@kaliA:~# export COMPARTIR_2='/media/loop1/loop1p2' #Declara unha variable de nome **COMPARTIR2** co valor **/media/loop1/loop1p2** e ademais exporta a variable para que poida ser recoñecida na contorna actual da shell.
root@kaliA:~# mount /dev/loop1p1 ${COMPARTIR_1} #Montar /dev/loop1p1 en /media/loop1/loop1p1
root@kaliA:~# mkdir ${COMPARTIR_1}/home #Crear cartafol /media/loop1/loop1p1/home
root@kaliA:~# mount /dev/loop1p2 ${COMPARTIR_2} #Montar /dev/loop1p2 en /media/loop1/loop1p2
root@kaliA:~# cp -Rav /home ${COMPARTIR_2} #Copiar de forma recursiva (-R) o contido do directorio /home en /media/loop1/loop1p2/home
root@kaliA:~# mount --bind ${COMPARTIR_2}/home ${COMPARTIR_1}/home #Montar grazas á opción --bind unha copia de /home accesible noutro lugar (/media/loop1/loop1p2/home)
root@kaliA:~# ls -ld ${COMPARTIR_1} #Listar soamente os permisos do cartafol /media/loop1/loop1p1, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
root@kaliA:~# ls -ld ${COMPARTIR_1}/home #Listar soamente os permisos do cartafol /media/loop1/loop1p1/home, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
root@kaliA:~# ls -l ${COMPARTIR_1}/home #Listar de forma extendida (tipo atopado, permisos, propietarios...) o contido do directorio /media/loop1/loop1p1/home

Podemos facer esta montaxe permanente engadindo unha entrada no ficheiro /etc/fstab:
/home /media/loop1/loop1p1/home none rw,bind 0 0

12. **Exemplo1. NFSv4: Compartición NFS sen Kerberos**

i. Desactivar seguridade **/etc/default/nfs-kernel-server**

#Modificar o arquivo de configuración /etc/default/nfs-kernel-server para desactivar a seguridade Kerberos: NEED_SVCGSSD="no"

root@kaliA:~# sed -i 's/NEED_SVCGSSD=""/NEED_SVCGSSD="no"/' /etc/default/nfs-kernel-server #Modificar o arquivo de configuración /etc/default/nfs-kernel-server para desactivar Kerberos.

ii. **Recargar configuración Servidor NFS**:
root@kaliA:~# /etc/init.d/nfs-kernel-server reload #Recargar a configuración do servidor NFS.
root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.

iii. Arquivo **/etc/exports** tipo:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
#Compartir a unha rede mediante NFSv4
/media/loop1/loop1p1/ 192.168.120.0/24(rw,sync,fsid=0,no_subtree_check)
/media/loop1/loop1p1/home 192.168.120.0/24(rw,sync,nohide,insecure,no_subtree_check)
```

Opcións montaxe cliente:

- Configurar a opción **crossmnt en /compartir** (pseudo filesystem raíz) equivale a opción **nohide en /compartir/kali** (directorio fillo do pseudo filesystem raíz). As 2 opcións son excluintes. Empregar soamente unha.
- **insecure:** permite aos clientes conectar a portos maiores de 1024. Esta opción non é insegura doutro xeito.

iv. Aplicar cambios do ficheiro /etc/exports:

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportandos mediante NFS. De momento, como non actualizamos os cambios de configuración, non temos ningún exportado.
root@kaliA:~# exportfs -av #Ler o ficheiro /etc/exports e exportar o indicado na súa configuración. A opción -a permite exportar e a opción -v é o modo verbose(detallado), é dicir, amosar máis información na saída de execución do comando.

Se a opción **-a** emprégase coa opción **-u** elimina a exportación de todos os sistemas de ficheiros que estaban exportados.
exportfs -auv

root@kaliA:~# showmount -e #Revisar os sistemas de ficheiros exportados mediante NFS. Agora si, unha vez recargada a configuración, podemos ver os sistemas de ficheiros exportados.

v. **Recargar configuración Servidor NFS**:
root@kaliA:~# /etc/init.d/nfs-kernel-server reload #Recargar a configuración do servidor NFS.
root@kaliA:~# /etc/init.d/nfs-kernel-server status #Comprobar o estado do servidor NFS.

vi. **Cliente NFSv4: Conexión ao servidor NFSv4**:

Activar idmapd daemon **/etc/default/nfs-common**

#Modificar o arquivo de configuración /etc/default/nfs-common para activar o demo idmapd necesario para NFSv4: NEED_IDMAPD=yes

root@kaliA:~# sed -i 's/NEED_IDMAPD=/NEED_IDMAPD="yes"/' /etc/default/nfs-common #Modificar o arquivo de configuración /etc/default/nfs-common para activar o demo idmapd

A. **EN TEMPO REAL**

- **mount -t nfs4**

mount -t nfs4 -o proto=tcp,port=2049 IP_Servidor:/ ruta_montaxe_local

mount -t nfs4 -o proto=tcp,port=2049 IP_Servidor:/cartafol_compartido ruta_montaxe_local

IMPORTANTE:
root@kaliB:~# mount -t nfs4 -o proto=tcp,port=2049 192.168.120.100:/ /mnt
O pseudo sistema de ficheiros raíz en NTFSv4 /media/loop1/loop1p1 será chamado a montar como / xa que a opción de exportación fsid permite ser a raíz de todos os puntos de montaxe a exportar.

root@kaliB:~# [! -d /mnt/home] && mkdir /mnt/home ; mount -t nfs4 -o proto=tcp,port=2049 192.168.120.100:/home /mnt/home

- **mount.nfs4**

mount.nfs4 -o proto=tcp,port=2049 IP_Servidor:/ ruta_montaxe_local

mount.nfs4 -o proto=tcp,port=2049 IP_Servidor:/cartafol_compartido ruta_montaxe_local

IMPORTANTE:
root@kaliB:~# mount.nfs4 -o proto=tcp,port=2049 192.168.120.100:/ /mnt
O pseudo sistema de ficheiros raíz en NTFSv4 /media/loop1/loop1p1 será chamado a montar como / xa que a opción de exportación fsid permite ser a raíz de todos os puntos de montaxe a exportar.

root@kaliB:~# [! -d /mnt/home] && mkdir /mnt/home ; mount.nfs4 -o proto=tcp,port=2049 192.168.120.100:/home /mnt/home

B. **DE FORMA PERMANENTE: /etc/fstab**

i. Engadir no ficheiro **/etc/fstab** a seguinte liña:

192.168.120.100:/ /mnt nfs4 _netdev,auto 0 0
192.168.120.100:/home /mnt/home nfs4 _netdev,auto 0 0

Opcións montaxe cliente:

- **_netdev** Montar sempre e cando a rede esté activa.
- **auto** Montar no arranque do sistema. Permitir montar coa opción -a

ii. Remontar sen ter que reiniciar tódolos puntos de montaxe definidos no **/etc/fstab** :

root@kaliB:~# mount -a

IMPORTANTE (NFSv4)

root@kaliB:~# mkdir /mnt/home/kali/crearDIR #Non se pode crear o directorio crearDIR sendo o usuario **root**.

root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

kali@kaliB:~$ mkdir /mnt/home/kali/crearDIR #Pódese crear o directorio crearDIR sendo o usuario **kali**.

Podemos comprobar que en NTFSv4

- Aínda que estamos a traballar co usuario **root** no equipo cliente, nos seus permisos **ugo** no directorio non posúe permisos de escritura e polo tanto non pode escribir nese directorio.
- O usuario kali si ten permisos de escritura e polo tanto SI pode escribir no directorio.
- O tipo de autenticación empregado é o do sistema, é dicir, emprga UID/GID nas conexións NFS.

  kali@kaliB:~$ cat /proc/mounts #Podemos revisar permisos de compartición NFS coa execución deste comando.

iii. Desmontar sistemas de ficheiros compartidos:

root@kaliB:~# umount /mnt/home /mnt

13. **Exemplo2. Compartición NFS sen Kerberos - TCP wrappers (/etc/hosts.allow, /etc/hosts.deny)**

    NOTA: É preciso ter realizado o Exemplo1.

    i. Comentar todas as liñas do arquivo **/etc/hosts.allow**:

        root@kaliA:~# sed -i 's/^/#/' /etc/hosts.allow

    ii. Engadir ao arquivo **/etc/hosts.deny**:

        ```
        ALL: 192.168.120.101
        ```

        root@kaliA:~# echo 'ALL: 192.168.120.101' >> /etc/hosts.deny

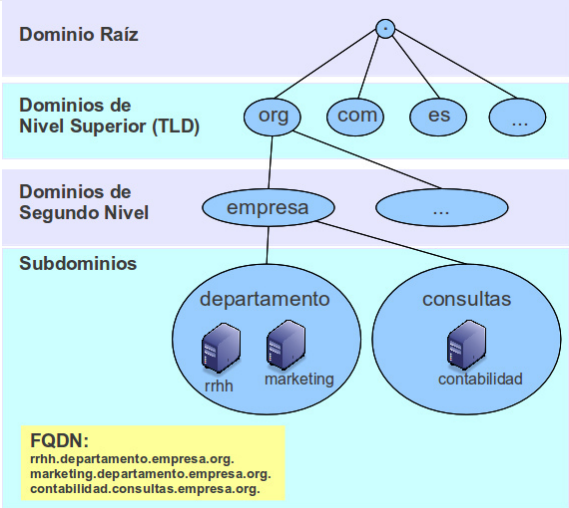
        Prioridade
        Primeiro lense os TCP wrappers (/etc/hosts.allow e /etc/hosts.deny) e se estes permiten o acceso será lido o ficheiro /etc/exports. Neste caso non temos nada configurado en /etc/hosts.allow co cal segue a lectura a /etc/hosts.deny, e aquí si que temos configurado que o host 192.168.120.101 ten denegado o acceso, polo tanto cando se cumpra esta condición o arquivo /etc/exports non se lerá.

    iii. Desmontar o recurso compartido e voltar a montalo:
        root@kaliA:~# umount /mnt/home /mnt #Desmontar os recursos compartidos
        root@kaliA:~# mount -a #Voltar a montalo realizando o apartado vi.B do Exemplo1.

        IMPORTANTE:
        Segue tendo lugar a compartición por rede mediante NFSv4, xa que non se cumpre a condición configurada en /etc/hosts.deny:
        ALL: 192.168.120.101
        Debido a que NFSv4 non depende de portamap(rpcbind), ou sexa, non depende da libraría libwrap de TCP wrappers:
            root@kaliA:~# ldd /usr/sbin/rpcbind | grep libwrap #Consultar as librarías dinámicas cargadas no comando rpcbind

Ricardo Feijoo Costa

DNS (Domain Name Server)



Nomenclatura DNS

DNS: O sistema DNS é unha base de datos distribuída, que permite a administración local de segmentos que xuntos compoñen toda a base de datos local. Os datos de cada segmento están dispoñibles para toda a rede a través dun esquema cliente-servidor xerárquico.

Servidor DNS: Permite resolver hostnames a IPs e viceversa.

/etc/hosts: Ficheiro que soe ter preferencia (ver `/etc/nsswitch.conf`) na resolución de nomes sobre o servidor DNS. Permite alias de nomes de dominios, é dicir, unha mesma IP pode apuntar a nomes distintios. Cada liña do ficheiro comezará con unha IP e na mesma liña separados por espazos ou tabuladores podes escribir os nomes de dominios correspondentes. O primeiro nome, o máis preto á IP, é considerado o principal, os demais son alias de éste.

/etc/nsswitch.conf: Ficheiro de configuración das Bases de Datos do Sistema e do sistema de Conmutación dos Servizos de Nomes (Name Service Switch - NSS), é dicir, establece a orde de búsqueda das bases de datos que ten definidas. A base de datos que corresponde á resolución DNS é hosts. Así, se neste ficheiro atopamos: `hosts: files dns` na resolución DNS prevalece o existente en /etc/hosts

Zona DNS: É aquela parte do DNS para a cal se delegou a administración. é dicir, cando se configura un dominio nun servidor DNS este debe pertencer a unha zona. Así, nos arquivos de configuración de zona indicarase que IP vai co servizo web www, o servizo de correo mail, etc.

Zona de Procura Directa: As resolucións desta zona devolven a dirección IP correspondente ao recurso solicitado. Realiza as resolucións que esperan como resposta a dirección IP dun determinado recurso.

Zona de Procura Inversa: As resolucións desta zona buscan un nome de equipo en función da súa dirección IP; unha procura inversa ten forma de pregunta, do estilo: Cal é o nome DNS do equipo que utiliza a dirección IP `192.168.100.10`?.

DNS Dinámico: É un sistema que permite a actualización en tempo real da información sobre nomes de dominio situados nun servidor de nomes, sendo usado maioritariamente para asignar un nome de dominio da internet a un computador con dirección IP variable (dinámica). O DNS dinámico, así, pode ofrecer servizos na internet en hosts que posúan conexión con dirección IP dinámica, a típica configuración que os ISP ofrecen para conectarse a Internet.

Servidores primarios (primary name servers): Estes servidores almacenan a información da súa zona nunha base de datos local. Son os responsables de manter a información actualizada e calquera cambio debe ser notificado a este servidor.

Servidores secundarios (secundary name servers): Tamén chamados escravos, aínda que á súa vez poden ser mestres doutros servidores secundarios. Son aqueles que obteñen os datos da súa zona desde outro servidor que teña autoridade para esa zona. O proceso de copia da información denomínase transferencia de zona.

Servidores mestres (master name servers): Os servidores mestres son os que transfiren as zonas aos servidores secundarios. Cando un servidor secundario arrinca busca un servidor mestre e realiza a transferencia de zona. Un servidor mestre para unha zona pode ser á vez un servidor primario ou secundario desa zona. Así, evítase que os servidores secundarios sobrecarguen ao servidor primario con transferencias de zonas. Os servidores mestres extraen a información desde o servidor primario da zona

Servidores só caché (caching-only servers): . Os servidores só caché non teñen autoridade sobre ningún dominio: limítanse a contactar con outros servidores para resolver as peticións dos clientes DNS. Estes servidores manteñen unha memoria caché coas últimas preguntas contestadas. Cada vez que un cliente DNS formúlalle unha pregunta, primeiro consulta na súa memoria caché. Se atopa a dirección IP solicitada, devólvella ao cliente; se non, consulta a outros servidores, apunta a resposta na súa memoria caché e comunícalle a resposta ao cliente. Se o noso caché DNS almacena a gran maioría de peticións que se realizan desde a rede local, as respostas dos clientes satisfaranse practicamente de forma instantánea proporcionando ao usuario unha sensación de velocidade na conexión. Todos os servidores DNS gardan na caché as consultas que resolveron.

Transferencia de zona: O proceso de copia da información de zonas entre servidores DNS denomínase transferencia de zona. Unha transferencia de zona pode darse: cando vence o intervalo de actualización dunha zona, cando un servidor mestre notifica os cambios da zona a un servidor secundario, cando se inicia o servizo Servidor DNS nun servidor secundario da zona, cando se utiliza o comando rndc nun servidor secundario da zona para iniciar manualmente unha transferencia desde o seu servidor mestre

Servidores raíz: Os servidores de raíz son entidades distintas. Hai 13 servidores raíz ou, máis precisamente, 13 direccións IP na internet nas que poden atoparse aos servidores raíz (os servidores que teñen unha das 13 direccións IP poden atoparse en ducias de localizacións físicas distintas). Todos estes servidores almacenan unha copia do mesmo arquivo que actúa como índice principal das axendas de direccións da internet. Enumeran unha dirección para cada dominio de nivel principal (.com, .es, etc.) na que pode atopase a propia axenda de direccións dese rexistro. Os trece servidores raíz DNS denomínanse polo primeiras trece letras do alfabeto latino, da **A ata a M (`A.ROOT-SERVERS.NET.`, `B.ROOT-SERVERS.NET.`, `...` , `M.ROOT-SERVERS.NET.`)**, e están en mans de 12 organizacións independentes.

Rexistros DNS: Cada zona DNS mantén un conxunto de rexistros de recursos (RR) estruturados.

Recursividade: Un servidor DNS tamén pode consultar ou poñerse en contacto con outros servidores DNS en nome do cliente DNS solicitante para resolver o nome por completo e, a continuación, enviar unha resposta ao cliente. Este proceso chámase recursividade.

Iteración: Un cliente DNS pode tentar poñerse en contacto con servidores DNS adicionais para resolver un nome. Cando un cliente fai isto, utiliza consultas adicionais e independentes en función de respostas de referencia dos servidores. Este proceso chámase iteración.

Xerarquía de nomes de dominio

O espazo de nomes de dominio (o universo de todos os nomes de dominio) está organizado de forma xerárquica. O nivel máis alto na xerarquía é o dominio raíz, que se representa como un punto (".") e o seguinte nivel na xerarquía chámase dominio de nivel superior (**TLD**). Só hai un dominio raíz, pero hai moitos TLDs e cada TLD chámase dominio secundario do dominio raíz. Neste contexto, o dominio raíz é o dominio principal, xa que está un nivel por encima dun TLD e cada TLD, á súa vez, poden ter moitos dominios fillos. Os fillos dos dominios de nivel superior chámanse de segundo nivel, os do segundo nivel chámanse de terceiro nivel, os do terceiro nivel de cuarto, e así sucesivamente.
Por tanto o DNS, organiza os nomes de máquina (hostname) nunha xerarquía de dominios separados polo carácter punto '.'. Un dominio é unha colección de nodos relacionados dalgunha forma -porque están na mesma rede, tal como os nodos dunha empresa-. Por exemplo:

rrhh.departamento.empresa.org
márketing.departamento.empresa.org
contabilidade.consultas.empresa.org

onde:

- A empresa agrupa as súas nodos no dominio de primeiro nivel `org`. Este é un TLD.
- A empresa ten un subdominio, dominio de segundo nivel `empresa` baixo `org`. Así `empresa` é un dominio de segundo nivel, fillo do TLD `org`.
- Á súa vez podes atopar novos subdominios dentro, neste caso: `departamento` e `consultas`. É dicir, dominios de terceiro nivel, fillos á súa vez do dominio de segundo nivel `empresa`.
- Finalmente, un nodo que terá un nome completo coñecido como totalmente cualificado ou FQDN, que é a concatenación de: TLD, dominio de segundo nivel, dominio de terceiro nivel, etc., tal como:
  - `rrhh.departamento.empresa.org`.
  - `márketing.departamento.empresa.org`.
  - `contabilidade.consultas.empresa.org`.

Na figura podes ver unha parte do espazo de nomes. A raíz da árbore, que se identifica cun punto sinxelo, é o que se denomina dominio raíz e é a orixe de todos os dominios. Para indicar que un nome é FQDN, ás veces termínase a súa escritura nun punto, aínda que polo xeral se omite. Este punto significa que o último compoñente do nome é o dominio raíz. Así, por exemplo no nome de dominio:

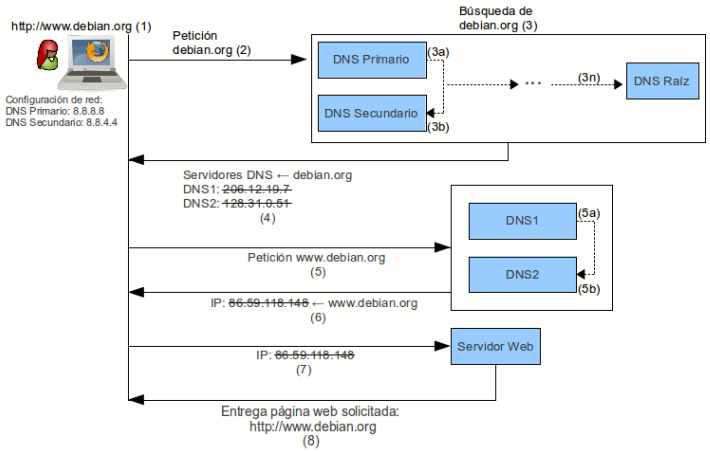
rrhh.departamento.empresa.org.

O símbolo do dominio raíz é o punto situado máis á dereita do nome do dominio.
Só hai unha raíz de dominio, pero **hai máis de 1500 dominios de nivel superior**, clasificados no seguintes tres tipos:

- TLD de código de país (ccTLD): dominios asociados con países e territorios. Hai máis de 240 ccTLD. Están formados por 2 letras, por exemplo: `es`, `uk`, `en`, e `jp`.
- Dominios de nivel superior xenéricos (gTLD): Están formados por 3 ou máis letras. Á súa vez se subdividen en:
  - Dominios da internet patrocinados (sTLD): Representan unha comunidade de intereses, é dicir, detrás existe unha organización ou organismo público que propón o dominio e establece as regras para obtar ao devandito dominio. Por exemplo: `edu`, `gov`, `int`, `mil`, `aero`, `museum`.
  - Dominios da internet non patrocinados (uTLD). Sen unha organización detrás que estableza as regras para obtar ao devandito dominio. A lista de gTLD inclúe: `com`, `net`, `org`, `biz`, `info`.

Funcionamento do DNS

A seguinte imaxe presenta graficamente o funcionamento do DNS, tomando como exemplo a páxina web `www.debian.org` e considerando que a información da petición do dominio para buscar non se atopa no teu computador ou nun servidor DNS local existente na túa rede ou no teu computador.



1. A través do teu navegador queres consultar a páxina web oficial de Debian escribindo na barra de direccións a URL `http://www.debian.org`.

2. O navegador busca a información das DNS do dominio `debian.org`.

3. Internet está ordenada en forma de árbore investida, se non atopa a información no teu computador, irá buscala aos servidores DNS que posúes na configuración de rede do teu computador, tipicamente os proporcionados polo teu Provedor de Servizos a Internet (ISP): `DNS Primario (3a)` ou `DNS Secundario (3b)`. De non estar, seguirá buscándoa a niveis superiores, e en último lugar atoparao no Servidor de Nomes Raíz: `DNS Raíz (3n)`.

4. A información buscada: as IP correspondentes ao servidor DNS que goberna o dominio `debian.org`, chega ao teu computador: `DNS1 →` ~~`206.12.19.7`~~ `e DNS2 →` ~~`128.31.0.51`~~. Adoitan ser dous porque as especificacións de deseño de DNS recomendan que como mínimo deben existir dous servidores DNS para aloxar cada zona, á que pertence cada dominio.

O teu computador agora tentará conectar co servidor `DNS1 (5a)` ou ante calquera problema de conexión con este tentarao co servidor `DNS2 (5b)`. Estes son os servidores de nomes onde se atopa información acerca de onde se pode buscar a páxina web (servidor da web), unha dirección de correo electrónico (servidor de correo), etc.

5. O teu computador recibirá a información acerca da localización da páxina web, ou sexa, a dirección IP do servidor web onde está aloxada a páxina.

6. O teu computador dirixirase logo ao servidor web e buscará a páxina web en éste.

7. Por último, o servidor web devolve a información pedida e ti recibes a páxina web, visualizándoa no navegador.

Pero, e se volves consultar a páxina web oficial de Debian escribindo na barra de direccións a URL `http://www.debian.org`, repetirase de novo todo o proceso? Para contestar este pregunta hai que establecer dúas situacións:

1. O host desde o que volves realizar a consulta é o mesmo: Se non o é, antes de repetir todo o proceso tentaríase co exposto no seguinte punto, pero se é o mesmo, ao facer a consulta desde este host, a resolución dominio-IP se garda durante algún tempo na memoria caché do mesmo, polo cal non será necesario repetir todo o proceso de novo. Se o tempo no que a memoria caché garda a resolución expirou volverá repetir o proceso de novo.

2. Existe un servidor DNS caché na túa rede ou no teu host: por tanto, se un segundo cliente, que ten configurado este servidor DNS, volve realizar a mesma petición, como este servidor ten a resposta almacenada na súa memoria caché, responderá inmediatamente sen ter que cursar a petición a ningún servidor DNS da internet. Se o tempo no que a memoria caché garda a resolución expirou volverá repetir o proceso de novo.

Tipos de rexistros DNS

Todos os rexistros de recursos (RR) teñen un formato definido que utiliza os mesmos campos de nivel superior, segundo descríbese na táboa seguinte:

Formato dos rexistros de recursos DNS	
Campo	**Descrición**
Propietario	Indica o nome de dominio DNS que posúe un rexistro de recursos. Este nome é o mesmo que o do nodo da árbore da consola onde se atopa un rexistro de recursos.
Tempo de vida (TTL)	Para a maior parte dos rexistros de recursos, este campo é *opcional*. Indica o espazo de tempo utilizado por outros servidores DNS para determinar canto tarda a información en caché en caducar un rexistro e descartalo. Por exemplo, a maior parte dos rexistros de recursos que crea o servizo do servidor DNS herdan o TTL mínimo (predeterminado) de 1 hora desde o rexistro de recurso de inicio de autoridade (SOA) que evita que outros servidores DNS almacenen en caché durante demasiado tempo. Nun rexistro de recursos individual, pode especificar un TTL específico para o rexistro que suplante o TTL mínimo (predeterminado) herdado do rexistro de recursos de inicio de autoridade. Tamén se pode utilizar o valor cero (0) para o TTL nos rexistros de recursos que conteñan datos volátiles que non estean na memoria caché para o seu uso posterior unha vez complétese a consulta DNS en curso.
Clase	Contén texto nemotécnico estándar que indica a clase do rexistro de recursos. Por exemplo, o valor "IN" indica que o rexistro de recursos pertence á clase Internet. Este campo é *obligatorio*.
Tipo	Contén texto nemotécnico estándar que indica o tipo de rexistro de recursos. Por exemplo, o texto nemotécnico "A" indica que o rexistro de recursos almacena información de direccións de host. Este campo é *obligatorio*.
Datos específicos do rexistro	Un campo de lonxitude variable e *obrigatorio* con información que describe o recurso. O formato desta información varía segundo o tipo e clase do rexistro de recursos.

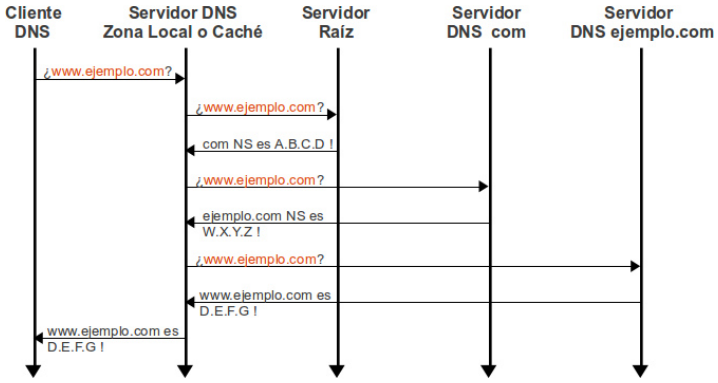
Tipos de rexistros DNS (O campo TTL *se omite en caso de ser opcional*. O campo TTL incluíuse na sintaxe de cada rexistro para indicar onde pode agregarse)	
Rexistro	**Descrición, Sintaxe e Exemplo**
A	**Descrición:** Address (Dirección). Este rexistro úsase para traducir nomes de hosts a direccións IP versión 4. **Sintaxe:** *propietario clase ttl* **A** *IP_version4* **Exemplo: host1.exemplo.com IN A 127.0.0.1**
AAAA	**Descrición:** Address (Dirección). Este rexistro úsase para traducir nomes de hosts a direccións IP versión 6. **Sintaxe:** *propietario clase ttl* **AAAA** *IP_version6* **Exemplo: host1ipv6.exemplo.com. IN AAAA 1234:0:1:2:3:4:567:89ab**
SRV	**Descrición:** Service Record (Rexistro de servizo). Este rexistro úsase para identificar os hosts que prestan servizos específicos. **Sintaxe:** *_servizo._proto.nome ttl clase* **SRV** *prioridade peso porto destino* **Exemplo: _ldap._tcp.exemplo.com. SRV 0 1 389 old-host1.exemplo.com** **_ldap._tcp.exemplo.com. SRV 0 5 389 new-host1.exemplo.com** Os servidores old-host1 e new-host1 teñen a mesma prioridade pero como new-host1 ten maior peso a conexión por defecto establecerase co servidor new-host1
CNAME	**Descrición:** Canonical Name (Nome Canónico). Úsase para crear nomes de hosts adicionais, ou alias. Hai que ter en conta que o nome de host ao que o alias referencia debe ser definido previamente como rexistro tipo "A". Comunmente usado cando un servidor cunha soa dirección IP executa varios servizos, como: ftp, web... e cada servizo ten a súa propia entrada DNS. Tamén é utilizado cando o servidor web aloxa distintos dominios nunha mesma IP (virtualhosts). **Sintaxe:** *propietario ttl clase* **CNAME** *nomeCanónico* **Exemplo: nomealias.exemplo.com CNAME nomeverdadeiro.exemplo.com** Como se comentou anteriormente **nomeverdadeiro.exemplo.com** previamente debe estar definido como rexistro tipo **A.**
NS	**Descrición:** Name Server (Servidor de Nomes). Indica que servidores de nomes teñen total autoridade sobre un dominio concreto. Cada dominio pódese asociar a unha cantidade calquera de servidores de nomes. **Sintaxe:** *propietario ttl IN* **NS** *nomeServidorNomeDominio* **Exemplo: exemplo.com. IN NS nomeservidor1.exemplo.com**
MX	**Descrición:** Mail eXchange (Rexistro de Intercambio de Correo). Asocia un nome de dominio a unha lista de servidores de intercambio de correo para ese dominio. **Sintaxe:** *propietario ttl clase* **MX** preferencia hostIntercambiadorDeCorreo **Exemplo: exemplo.com. MX 10 servidorcorreo1.exemplo.com** O número, neste caso 10, indica a preferencia, e ten sentido en caso de existir varios servidores de correo. A menor número maior preferencia.
PTR	**Descrición:** Pointer (Indicador). Traduce direccións IP en nomes de dominio. Tamén coñecido como 'rexistro inverso', xa que funciona á inversa do rexistro "A". **Sintaxe:** *propietario ttl clase* **PTR** *nomeDominioDestino* **Exemplo: 1.0.0.10.in-addr.arpa. PTR host.exemplo.com**
SOA	**Descrición:** Start Of Authority (Autoridade da zona). Proporciona información sobre o servidor DNS primario da zona. **Sintaxe:** *propietario clase servidorNomes persoaResponsable (numeroSerie intervaloActualización intervaloReintento caducidadetempoDeVidaMínimo)* **Exemplo: @ IN SOA nomeServidor.exemplo.com. postmaster.exemplo.com. (** **1 ; número de serie** **3600 ; actualizar [1h]** **600 ; reintentar [10m]** **86400 ; caducar [1d]** **3600) ; TTL mínimo [1h]** O propietario (servidor DNS principal) especifícase como "@" porque o nome de dominio é o mesmo que a orixe de todos os datos da zona (exemplo.com.). Trátase dunha convención de nomenclatura estándar para rexistros de recursos e utilízase máis a miúdo nos rexistros SOA. O número de serie é o número de versión desta base de datos. Debes incrementar este número cada vez que modificas a base de datos.
TXT	**Descrición:** TeXT (Información textual). Permite aos dominios identificarse de modos arbitrarios. **Sintaxe:** *propietario ttl clase* **TXT** *cadenaDeTexto* **Exemplo: exemplo.com. TXT "Exemplo de información de nome de dominio adicional."**
SPF	**Descrición:** Sender Policy Framework. É un rexistro de tipo TXT que vai creado nunha zona directa do DNS, na cal se pon as informacións do propio servidor de correo coa sintaxe SPF. Utilízase para evitar o envío de correos suplantando identidades. Por tanto, axuda a combater o SPAM, xa que, neste rexistro especifícase cal ou cales hosts están autorizados a enviar correo desde o dominio dado. O servidor que recibe, consulta o SPF para comparar a IP desde a cal lle chega, cos datos deste rexistro. **Sintaxe:** *propietario ttl clase* **IN SPF** *cadenaDeTexto* **Exemplo: exemplo.com IN SPF "v=spf1 a:mail.exemplo.com -all"**

Funcionamento do cliente DNS

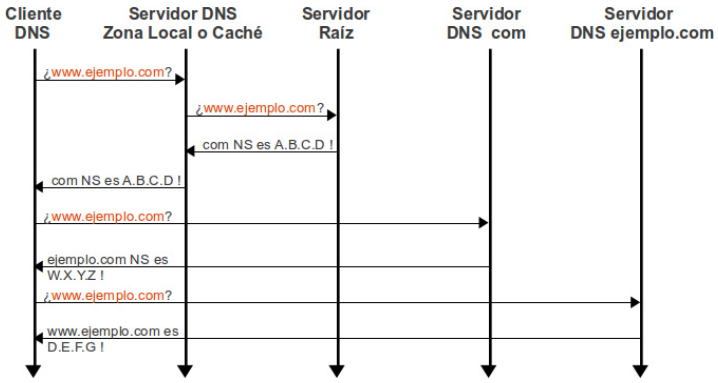
O proceso de consulta DNS realízase en dous partes:

- A consulta dun nome comeza nun equipo cliente e pásase ao solucionador (resolver), o servizo Cliente DNS, para proceder á súa resolución.
- Cando a consulta non se pode resolver localmente, pódese consultar aos servidores DNS segundo sexa necesario para resolver o nome.

Consultas recursivas

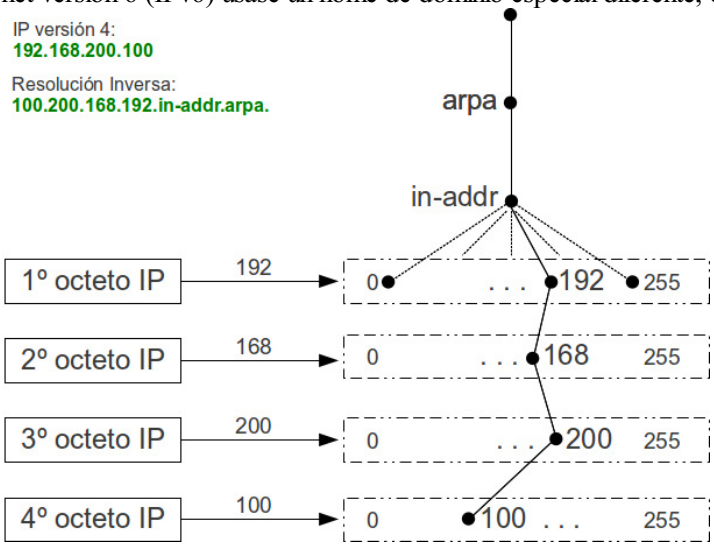


Consultas iterativas



Consultas inversas

O dominio `in-addr.arpa` úsase en todas as redes TCP/IP que se basean no direccionamiento do Protocolo da internet versión 4 (IPv4). Para o Protocolo da internet versión 6 (IPv6) úsase un nome de dominio especial diferente, o dominio `ip6.arpa`.



Ricardo Feijoo Costa

dnsmasq: Servizos DNS + DHCP

ESCENARIO

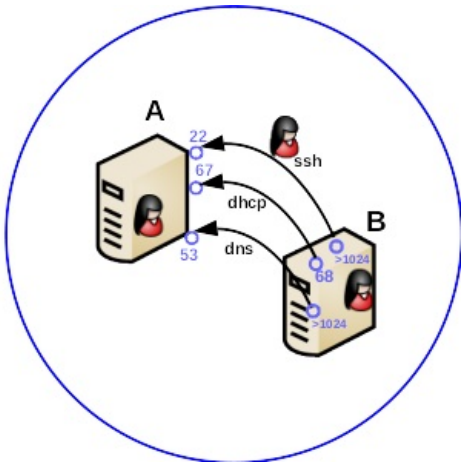
Máquinas virtuais:
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado
Rede: 192.168.120.0
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede Interna e NAT
Servidor SSH: openssh-server
Servidor DNS+DHCP: dnsmasq

ISO: Kali Live amd64
IP/MS: 192.168.120.100/24

Máquina virtual B:
Rede Interna
Cliente SSH: openssh-client (ssh)
Resolvers DNS: nslookup, dig, hosts
Cliente DHCP: dhclient

ISO: Kali Live amd64
IP/MS: 192.168.120.101/24



NOTAS:

- **dnsmasq** integra un sinxelo servidor DNS e servidor DHCP/TFTP fáciles de implementar. Paquete dnsmasq (# apt update && apt -y install dnsmasq)
- Ficheiros de configuración dnsmasq:
  - Servidor DNS: **/etc/hosts**
  - Servidor DHCP: **/etc/dnsmasq.conf**
- Ficheiro de configuración **/etc/resolv.conf**: Arquivo onde se configuran os servidores DNS que solucionan as peticións de nomes directa ou inversa.
- Cliente DHCP: Comando **dhclient**. Paquete isc-dhcp-client (# apt update && apt -y install isc-dhcp-client).
- Clientes DNS:
  - Comandos **nslookup** e **dig**. Paquete bind9-dnsutils (#apt update && apt -y install bind9-dnsutils)
  - Comando **host**. Paquete bind9-host (#apt update && apt -y install bind9-host)
- Cliente SSH: Comando ssh. Paquete openssh-client (# apt update && apt -y install openssh-client).
- Servidor SSH: Paquete openssh-server (# apt update && apt -y install openssh-server).

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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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** ^{SSH} **➜ 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:~$

Máquina virtual A: Kali amd64

8. **Instalar dnsmasq (DNS + DHCP)**:

kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@kaliA:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)

root@kaliA:~# apt search dnsmasq #Buscar calquera paquete que coincida co patrón de búsqueda dnsmasq

root@kaliA:~# apt -y install dnsmasq #Instalar o paquete dnsmasq. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

9. **Habilitar servizo dnsmasq(DNS + DHCP)**:

root@kaliA:~# /etc/init.d/dnsmasq status #Comprobar o estado do servidor dnsmasq

root@kaliA:~# /etc/init.d/dnsmasq start #Arrancar o servidor dnsmasq.

root@kaliA:~# /etc/init.d/dnsmasq status #Comprobar o estado do servidor dnsmasq

10. **dnsmasq como servidor DNS**

a. **/etc/resolv.conf e /etc/hosts**:

Para que dnsmasq poida realizar a resolución directa ou inversa de hostnames/IPs simplemente temos que engadir/modificar entradas no ficheiro /etc/hosts e indicarlle ao arquivo /etc/resolv.conf cal é o noso servidor DNS a empregar.

/etc/resolv.conf: Arquivo onde se configuran os servidores DNS. Exemplo contido tipo:

domain example.local #Dominio a engadir na procura de hostnames. Se o host a buscar é pepito, é a procura falla, intentariase de novo esta como pepito.example.local

search example.local #Lista de dominios a engadir na procura de hostnames.

nameserver 8.8.8.8 #Servidor DNS primario para resolución de nomes.

nameserver 8.8.4.4 #Agregar servidor DNS secundario para resolución de nomes.

domain e search son excluintes, a última directiva que apareza no ficheiro prevalece.

root@kaliA:~# echo -e 'nameserver 127.0.0.1\nnameserver 192.168.120.100' > /etc/resolv.conf #Agregar servidor DNS para resolución de nomes.

root@kaliA:~# echo '192.168.120.100 kaliA.ies.local kaliA.ies.com kaliA.example.local kaliA.example.gl' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) varios nomes DNS que apuntan á IP 192.168.120.100

root@kaliA:~# echo '192.168.120.101 kaliB.ies.local kaliB.ies.com kaliB.example.local kaliB.example.gl' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) varios nomes DNS que apuntan á IP 192.168.120.101

b. **Comprobación resolución DNS: DNS en /etc/hosts**

Á hora de saber se tes conectividade con algunha máquina na internet, ou en rede local, adóitase utilizar o comando ping, o cal indica segundo a súa resposta se posúes conectividade coa máquina en cuestión. O comando ping pódelo utilizar para consultar direccións IP ou nomes de dominios. Por tanto o comando ping debe ser capaz de consultar información sobre o sistema de nomes de dominio; é un resolutor, un programa cliente capaz de consultar información sobre o sistema de nome de dominios. Normalmente, un resolutor traballa discretamente en segundo plano e os usuarios non coñecen a súa presenza, é dicir, que toda consulta dun cliente DNS ao seu servidor adoita realizala o programa que invocamos (ping, ftp, telnet, mail, navegador web, etc.). Por exemplo, se solicitas unha conexión ftp a ftp. rediris.es, a aplicación ftp que empregues chama a un programa resolutor local que busca a dirección IP dese computador 130.206.13.2 sen que teñas conciencia diso, isto é, para ti o proceso é transparente. Ademais deste traballo en segundo plano, o usuario pode conectarse directamente ao programa resolutor enviando consultas e resolvendo respostas. Comandos resolutor típicos en sistemas operativos GNU/Linux son: nslookup, host e dig.

root@kaliA:~# ping -c4 kaliA.ies.local #Comprobar mediante o localando ping a conectividade co host kaliA.ies.local que apunta á interface de rede da máquina virtual A

root@kaliA:~# ping -c4 kaliA.ies.com #Comprobar mediante o comando ping a conectividade co host kaliA.ies.com que apunta á interface de rede da máquina virtual A

root@kaliA:~# ping -c4 kaliA.example.local #Comprobar mediante o comando ping a conectividade co host kaliA.example.local que apunta á interface de rede da máquina virtual A
root@kaliA:~# nslookup kaliA.example.gl #Resolución directa: Consultar a dirección IP do host kaliA.example.gl
root@kaliA:~# host kaliA.ies.com #Comando equivalente ao anterior
root@kaliA:~# dig kaliA.ies.com #Comando equivalente ao anterior
root@kaliA:~# nslookup 192.168.120.100 #Resolución inversa: Consultar o nome do host que posúe a dirección IP 192.168.120.100
root@kaliA:~# host 192.168.120.100 #Comando equivalente ao anterior
root@kaliA:~# dig -x 192.168.120.100 #Comando equivalente ao anterior

Máquina virtual B: Kali amd64

11. **Apuntar ao DNS (dnsmasq en kaliA)**. Modificar **/etc/resolv.conf** e **/etc/hosts**:

    a. **/etc/resolv.conf e /etc/hosts**:

    Para que poidamos empregar dnsmasq como servidor de nomes, e así poida realizar a resolución directa ou inversa de hostnames/IPs, simplemente temos que indicarlle ao arquivo /etc/resolv.conf de KaliB cal é o noso servidor DNS a empregar.

    root@kaliA:~# exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.
    kali@kaliA:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.
    kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
    root@kaliB:~# echo 'nameserver 192.168.120.100' > /etc/resolv.conf #Agregar servidor DNS para resolución de nomes.

    b. **Comprobación resolución DNS**:

    root@kaliB:~# ping -c4 kaliA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
    root@kaliB:~# ping -c4 kaliA.ies.local #Comprobar mediante o localando ping a conectividade co host kaliA.ies.local que apunta á interface de rede da máquina virtual A
    root@kaliB:~# ping -c4 kaliA.ies.com #Comprobar mediante o comando ping a conectividade co host kaliA.ies.com que apunta á interface de rede da máquina virtual A
    root@kaliB:~# ping -c4 kaliA.example.local #Comprobar mediante o comando ping a conectividade co host kaliA.example.local que apunta á interface de rede da máquina virtual A
    root@kaliB:~# nslookup kaliA.example.gl #Resolución directa: Consultar a dirección IP do host kaliA.example.gl
    root@kaliB:~# host kaliA.ies.com #Comando equivalente ao anterior
    root@kaliB:~# dig kaliA.ies.com #Comando equivalente ao anterior
    root@kaliB:~# nslookup 192.168.120.100 #Resolución inversa: Consultar o nome do host que posúe a dirección IP 192.168.120.100
    root@kaliB:~# host 192.168.120.100 #Comando equivalente ao anterior
    root@kaliB:~# dig -x 192.168.120.100 #Comando equivalente ao anterior

Máquina virtual A: Kali amd64

12. **dnsmasq como servidor DHCP ➜ /etc/dnsmasq.conf**

Por defecto dnsmasq non activa o servidor DHCP. Entón para activalo debemos editar o arquivo de configuración /etc/dnsmasq.conf e configurar un pool de IPs a servir, isto é, configurar os rangos de IPs que queremos conceder para poder distribuir unha IP nunha solicitude dun cliente DHCP.

```
root@kaliA:~# echo 'dhcp-range=192.168.120.50,192.168.120.80,12h' >>
/etc/dnsmasq.conf #Crear rango de IPs a distribuir [50-80]. A concesión durará 12 horas.
root@kaliA:~# /etc/init.d/dnsmasq restart #Reiniciar o servidor dnsmasq.
root@kaliA:~# /etc/init.d/dnsmasq status #Comprobar o estado do servidor dnsmasq
```

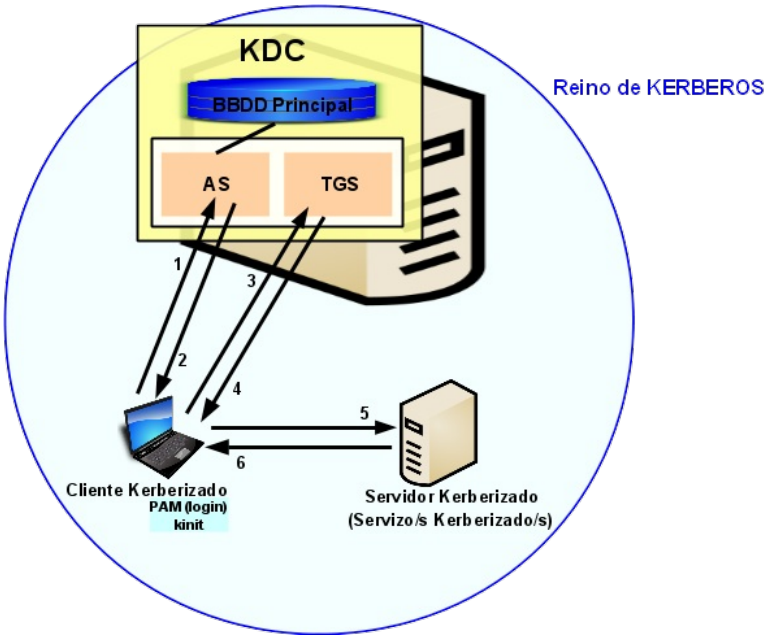
Máquina virtual B: Kali amd64

13. **dnsmasq como servidor DHCP ➜ Solicitar IP a dnsmasq**

```
root@kaliB:~# dhclient -v eth0 #Configuración dinámica de rede da interface eth0 en modo
verbose(detallado).
root@kaliB:~# ip addr show eth0 #Amosara configuración da interface eth0
root@kaliB:~# dhclient -s 192.168.120.100 -v eth0 #Configuración dinámica de rede da interface
eth0 en modo verbose(detallado), procurando a configuración no servidor DHCP 192.168.120.100
root@kaliB:~# ip addr show eth0 #Amosara configuración da interface eth0
```

Ricardo Feijoo Costa

Kerberos (autenticación, SSO, cifrado simétrico, KDC)



Kerberos MIT **Kerberos Heimdal** **Official documentation Ubuntu Kerberos**

kadmin
Comando para administrar Kerberos
de forma local ou remota

krb5-config
Comando para configuración

/etc/krb5.conf
Ficheiro de configuración

kinit
Comando que permite solicitar un ticket TGT
tras verificación de autententicación do usuario que procesa a solicitude.

NOMENCLATURA KERBEROS

KDC: Key Distribution Center → Emite tickets Kerberos
Ticket: Credenciais electrónicas temporais que verifican a identidade dun cliente para un servizo particular
BBDD Principal: Base de datos de usuarios do KDC ≠ /etc/passwd
AS: Authentication Server → Emite tickets de acceso TGT para gañar acceso ao servidor TGS, é dicir, encárgase de validar ao usuario frente ao sistema sustituíndo ao login clásico
TGS: Ticket Granting Server → Emite tickets para un servizo desexado, os cales son entregados aos usuarios para que poidan acceder ao servizo
TGT: Ticket especial que permite ao cliente obter tickets sen solicitalos dende o KDC
Cliente Kerberizado (PAM(login), kinit ...)
Servidor Kerberizado (Servizo/s Kerberizado/s)
Key: Chave → Datos usados cando ciframos ou desciframos datos. Non se pode descifrar os datos cifrados sen a chave correcta
Reino de Kerberos (Realm): Rede que usa Kerberos, composto por un ou varios servidores KDCs e un número potencial de clientes. O nome do reino de Kerberos para o KDC de Microsoft é o nome do controlador de dominio en MAIÚSCULAS, por exemplo: EDUCACION.LOCAL . Un cliente en Kerberos identifícase co seu principal
Principal (nome do principal): É o nome único do usuario ou servizo que pode autenticar mediante o uso de Kerberos.
Formato: name[/instance]@REALM
→ name para os usuarios é o mesmo que o ID de inicio de sesión, por exemplo root
→ instance pode ser opcional no caso dos usuarios pero é obrigatorio para os servizos, por exemplo: alumno, alumno/admin, alumno/host1.educacion.local
→ REALM é o nome do reino de Kerberos, por exemplo EDUCACION.LOCAL . Todos os principais dun reino teñen a súa propia chave, sendo para os usuarios derivada do seu contrasinal e para os servizos xerada aleatoriamente.
Keytab: Para os servizos dun host é similar ao contrasinal dun usuario. Cada host que proporciona un servizo debe ter un arquivo local keytab, que contén o principal para o servizo en cuestión, o cal denomínase clave de servizo.

KERBEROS

Kerberos proporciona autenticación SSO (Single Sing-On), de xeito que un usuario soamente ten que autenticarse unha vez en cada sesión e pode facer uso de esta autenticación para tódolos servizos e equipos do reino de Kerberos.
Kerberos permite:

- Ao cliente probar a súa identidade ante un servidor
- Ao servidor probar a súa identidade frente aos clientes
- Unha vez autenticados, cifrar a comunicación
- Que en ningún momento o contrasinal do usuario sexa enviado por rede
- Que os contrasinais dos usuarios non estén almacenados nos equipos clientes
- Que os contrasinais dos usuarios estén almacenados cifrados no KDC, na BBDD Principal
- Que os tickets soamente poden usarse durante un tempo limitado. Debido a isto é necesario a sincronización temporal do KDC, cliente e servidores kerberizados.

Microsoft Active Directory emprega Kerberos como mecanismo de seguridade predeterminado. Cando se engaden usuarios a Microsoft Active Directory, a súa identificación de Windows é equivalente a un nome principal Kerberos.

IMPORTANTE:

- **Active directory** ➜ Non distingue entre maiúsculas e minúsculas os nomes dos servizos
- **Kerberos** ➜ Distingue entre maiúsculas e minúsculas os nomes dos servizos
- **Convencións:**:
  - Os reinos de Kerberos e os dominios de Active Directory están escritos con maiúscula.
  - Os nomes de host escríbense en minúscula.
  - As buscas de bases de datos distinguen entre maiúsculas e minúsculas.

EXPLICACIÓN PROTOCOLO KERBEROS

Autenticación en 3 fases:

Fase 1: Pasos 1 e 2. Autenticación de usuario
Fase 2: Pasos 3 e 4. Autorización de tipo de servizo
Fase 3: Pasos 5 e 6. Autorización dun servidor
A **Fase 1** soamente ten lugar unha vez, sendo a **Fase 2** e a **Fase 3** as que se repiten para acceder a múltiples servizos kerberizados.

- **Paso 1:** O usuario introduce credenciais (PAM(login, su ...) no cliente e envía o principal ao servizo AS do servidor KDC.
- **Paso 2:** O servizo AS do KDC verifica as credenciais na BBDD Principal e logo de autenticar envía ao cliente un TGT.
- **Paso 3:** O cliente quere acceder a un servizo polo que envía o TGT ao TGS do KDC.
- **Paso 4:** O TGS verifica o TGT e envía un ticket de servizo para o servizo ou aplicación destino.
- **Paso 5:** O cliente envía o ticket de servizo ao servizo kerberizado para a súa autenticación.
- **Paso 6:** Se o servizo kerberizado acepta o ticket establécese un contexto de seguridade e entón a aplicación do usuario pode intercambiar datos co servizo destino.

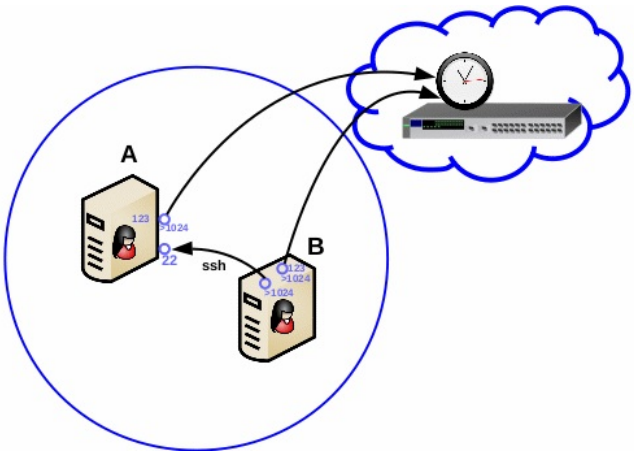
Finalmente, o cliente autenticouse (usuario/contrasinal) contra a BBDD Principal (Base de datos de usuarios do KDC ≠ /etc/passwd) → o cliente conta durante un período de tempo con un ticket TGT que permitira acceder a múltiples servizos kerberizados.

ESCENARIO

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

Máquina virtual A:	**Máquina virtual B:**
Rede Interna e NAT	Rede Interna e NAT
Servidor Kerberos: heimdal-kdc	Cliente Kerberos: heimdal-clients
Servidor SSH: openssh-server	Cliente SSH: openssh-client (ssh)
Servizo NTP: ntp	Servizo NTP: ntp
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



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:

   OPCIÓN A:
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliA || hostnamectl set-hostname kaliA #Modificar o hostname do sistema a kaliA.
   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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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

1. 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 || systemctl stop avahi-daemon #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 || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) 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

2. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

   OPCIÓN A:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   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**.

   OPCIÓN B:
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# hostnamectl hostname kaliB || hostnamectl set-hostname kaliB #Modificar o hostname do sistema a kaliB.
   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. **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:~$

NTP (sincronizar hosts para validez de tickets Kerberos)

Realizar nas 2 máquinas virtuais: A e B

Opción 1: Servizo NTP (ntpsec)

1. Instalar servizo NTP:
   # apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   # apt search ntp || apt-cache search ntp #Buscar calquera paquete que coincida co patrón de búsqueda ntp
   # apt -y install ntp || apt-get -y install ntp #Instalar o paquete ntp, é dicir, instalar o servizo NTP. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

2. Configurar servizo NTP (/etc/ntpsec/ntp.conf):
   # sed -i 's/debian.pool.ntp.org/es.pool.ntp.org/g' /etc/ntpsec/ntp.conf #Cambiar os servidores ntp cos que sincronizar o sistema (ver **http://www.pool.ntp.org/zone/es)**
   # systemctl restart ntp.service #Reiniciar o servizo ntp para ter en conta o cambio dos servidores realizado
   # nc -uvz localhost 123 #Mediante o comando nc(netcat) comprobar se o porto 123 do servizo NTP está activo. A opción -u indica que o porto a buscar emprega o protocolo UDP, 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 123 é o porto UDP a escanear.
   # ntpq -p #Identificar con que servidores ntp estamos a sincronizar o sistema

Opción 2: Servizo NTP (systemd-timesyncd)

3. Purgar servizo NTP (ntpd):
   # apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   # apt -y purge ntp || apt-get -y purge ntp #Eliminar o servizo NTP mediante a purga do paquete de nome *ntp*. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na desinstalación do paquete. **IMPORTANTE:** Con **purge SI SE ELIMINAN os ficheiros de configuración** do paquete desinstalado.
   # systemctl status systemd-timesyncd #Comprobar o estado do servizo systemd-timesyncd.

4. Configurar NTP mediante timesyncd (systemd, timedatectl):
   # A=$(grep -n 'NTP=' /etc/systemd/timesyncd.conf | cut -d':' -f1 | xargs | awk '{print $NF}')
   # sed -i -e 's/NTP=/##NTP=/g' -e
   "${A}a\NTP=2.es.pool.ntp.org\nNTP=3.europe.pool.ntp.org\nNTP=1.europe.pool.ntp.org"
   /etc/systemd/timesyncd.conf #Cambiar os servidores ntp cos que sincronizar o sistema (ver **http://www.pool.ntp.org/zone/es)**
   # systemctl restart systemd-timesyncd #Reiniciar o servizo ntp para ter en conta o cambio dos servidores realizado
   # timedatectl set-timezone Europe/Madrid #Modificar a zona temporal a Europe/Madrid

   $ timedatectl #Comando que controla a hora e data do sistema. Executado Sen opcións amosa información sobre como está configurada a hora/data do sistema (sincronización NTP, zona temporal...)
   $ timedatectl list-timezones #Lista as zonas temporais

dnsmasq: Servizos DNS + DHCP

dnsmasq (DNS + DHCP) na máquina virtual A

Convencións::

- Os reinos de Kerberos e os dominios de Active Directory están escritos con maiúscula.
- Os nomes de host escríbense en minúscula.
- As buscas de bases de datos distinguen entre maiúsculas e minúsculas.

1. Hostname. Pór kalia.ies.local como hostname:
   root@kaliA:~# echo 'kalia.ies.local' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kaliA:~# sed -i 's/kaliA/kalia.ies.local/' /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kaliA:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

2. Instalar dnsmasq:
   root@kalia:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   root@kalia:~# apt search dnsmasq #Buscar calquera paquete que coincida co patrón de búsqueda dnsmasq
   root@kalia:~# apt -y install dnsmasq #Instalar o paquete dnsmasq. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

3. Resolución DNS. Actualizar o arquivo /etc/hosts:
   root@kalia:~# echo '192.168.120.100 kalia.ies.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kalia.ies.local para que atenda á IP 192.168.120.100
   root@kalia:~# echo '192.168.120.101 kalib.ies.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kalib.ies.local para que atenda á IP 192.168.120.101
   root@kalia:~# hostname -f #Comprobar que se responde ao nome FQDN: kaliA.ies.local

4. Configurar /etc/resolv.conf para apuntar a dnsmasq como servidor DNS a empregar para as resolucións de nomes/IPs:

   /etc/resolv.conf: Arquivo onde se configuran os servidores DNS. Exemplo contido tipo:

   domain example.local #Dominio a engadir na procura de hostnames. Se o host a buscar é pepito, é a procura falla, intentariase de novo esta como pepito.example.local

   search example.local #Lista de dominios a engadir na procura de hostnames.

   nameserver 8.8.8.8 #Servidor DNS primario para resolución de nomes.

   nameserver 8.8.4.4 #Agregar servidor DNS secundario para resolución de nomes.

   domain e search son excluintes, a última directiva que apareza no ficheiro prevalece.

   root@kalia:~# echo -e 'nameserver 127.0.0.1\nnameserver 192.168.120.100\nnameserver 8.8.4.4' > /etc/resolv.conf #Agregar servidor DNS para resolución de nomes.

5. Habilitar servizo dnsmasq(DNS + DHCP):
   root@kalia:~# /etc/init.d/dnsmasq status #Comprobar o estado do servidor dnsmasq
   root@kalia:~# /etc/init.d/dnsmasq start #Arrancar o servidor dnsmasq.
   root@kalia:~# /etc/init.d/dnsmasq status #Comprobar o estado do servidor dnsmasq

Kerberos Heimdal

Servidor Kerberos Heimdal (kdc) na máquina virtual A

1. Instalar servidor Kerberos Heimdal e definir REALM:
   root@kalia:~# apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   root@kalia:~# apt search heimdal-kdc || apt-cache search heimdal-kdc #Buscar calquera paquete que coincida co patrón de búsqueda heimdal-kdc
   root@kalia:~# apt -y install heimdal-kdc || apt-get -y install heimdal-kdc #Instalar o paquete heimdal-kdc, é dicir, instalar o servidor Kerberos. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

   ```
   Default Kerberos version 5 realm:
   IES.LOCAL #Indicar reino Kerberos (MAIÚSCULAS)
   Kerberos servers for your realm:
   kalia.ies.local #Indicar servidor Kerberos (minúsculas)
   Administrative server for your Kerberos realm:
   kalia.ies.local #Indicar servidor administrador do reino Kerberos (minúsculas)
   ```

   ```
   root@kalia:~# dpkg-reconfigure kbr5-config #Reconfigurar kerberos
   root@kalia:~# apt -y install heimdal-docs #Instalar paquete documentación Heimdal Kerberos
   root@kalia:~# dpkg -L heimdal-docs #Listar ficheiros pertencentes ao paquete heimdal-docs
   ```

2. Comprobar estado do Servidor Kerberos:
   root@kalia:~# /etc/init.d/heimdal-kdc status #Comprobar o estado do servidor Kerberos, por defecto (logo de instalar o paquete) está arrancado.
   root@kalia:~# nc -vz localhost 88 #Mediante o comando nc(netcat) comprobar se o porto 88 do servidor Kerberos 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 88 é o porto TCP a escanear.
   root@kalia:~# nc -vz 192.168.120.100 88 #Mediante o comando nc(netcat) comprobar se o porto 88 do servidor Kerberos 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 88 é o porto TCP a escanear.
   root@kalia:~# nc -vz kalia.ies.local 88 #Mediante o comando nc(netcat) comprobar se o porto 88 do servidor Kerberos 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 88 é o porto TCP a escanear.
   root@kalia:~# netstat -natp | grep 88 #Mediante o comando netstat comprobar que o porto 88 do servidor Kerberos 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 88 #Mediante o comando ss comprobar que o porto 88 do servidor Kerberos 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.

3. Usuario e ticket. Autenticar como usuario en Kerberos, conseguir un ticket e ver a súa validez:
   root@kalia:~# kadmin -l #Administrar de forma local kerberos

   ```
   root@kalia:~# kadmin -l
   kadmin> add user1 #Crear usuario user1 na bddd principal kerberos (base de datos de usuarios do KDC)
   Max ticket life [1 day]:
   Max renewable life [1 week]:
   Principal expiration time [never]:
   Password expiration time [never]:
   Attributes []:
   Policy [default]:
   user1@IES.LOCAL's Password: #Introducir o contrasinal que queremos que posúa o usuario user1. Por exemplo abc123. (Ollo que o contrasinal ten un caracter punto final)
   Verify password - user1@IES.LOCAL's Password: #Introducir de novo o contrasinal anterior (abc123.)
   kadmin> quit
   root@kalia:~#
   ```

   Dentro de kadmin coa comando **list** * listamos todos os principales.

   root@kalia:~# kinit user1 #Autenticar como usuario **user1** e contrasinal **abc123.** conseguindo un ticket TGT
   user1@IES.LOCAL's Password:
   root@kalia:~# klist -l #Listar a validez do ticket TGT. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

Apuntar ao DNS (dnsmasq en kaliA)

1. Hostname. Pór kalib.ies.local como hostname:
   root@kaliB:~# echo 'kalib.ies.local' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kaliB:~# sed -i 's/kaliB/kalib.ies.local/' /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kaliB:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

2. Resolución DNS. Actualizar o arquivo /etc/hosts:
   root@kalib:~# echo '192.168.120.101 kalib.ies.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kalib.ies.local para que atenda á IP 192.168.120.101
   root@kalib:~# hostname -f #Comprobar que se responde ao nome FQDN: kalib.ies.local
   root@kalib:~# sed -i 's/kaliA/kalia.ies.local/' /etc/hosts #Modificar a entrada de kaliA no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) para que o nome kalia.ies.local atenda á IP 192.168.120.100

3. Configurar /etc/resolv.conf para apuntar a dnsmasq como servidor DNS a empregar para as resolucións de nomes/IPs:

   /etc/resolv.conf: Arquivo onde se configuran os servidores DNS. Exemplo contido tipo:

   domain example.local #Dominio a engadir na procura de hostnames. Se o host a buscar é pepito, é a procura falla, intentariase de novo esta como pepito.example.local

   search example.local #Lista de dominios a engadir na procura de hostnames.

   nameserver 8.8.8.8 #Servidor DNS primario para resolución de nomes.

   nameserver 8.8.4.4 #Agregar servidor DNS secundario para resolución de nomes.

   domain e search son excluintes, a última directiva que apareza no ficheiro prevalece.

   root@kalib:~# echo -e 'nameserver 192.168.120.100\nnameserver 8.8.4.4' > /etc/resolv.conf #Agregar servidor DNS para resolución de nomes.

Cliente Kerberos Heimdal na máquina virtual B

1. Instalar cliente Kerberos Heimdal e reino REALM:
   root@kalib:~# apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   root@kalib:~# apt search heimdal-clients || apt-cache search heimdal-clients #Buscar calquera paquete que coincida co patrón de búsqueda heimdal-clients
   root@kalib:~# apt -y install heimdal-clients || apt-get -y install heimdal-clients #Instalar o paquete heimdal-clients, é dicir, instalar o cliente Kerberos. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

   ```
   Default Kerberos version 5 realm:
   IES.LOCAL #Indicar reino Kerberos (MAIÚSCULAS)
   Kerberos servers for your realm:
   kalia.ies.local #Indicar servidor Kerberos (minúsculas)
   Administrative server for your Kerberos realm:
   kalia.ies.local #Indicar servidor administrador do reino Kerberos (minúsculas)
   ```

   ```
   root@kalib:~# dpkg-reconfigure kbr5-config #Reconfigurar kerberos
   ```

2. Usuario e ticket. Autenticar como usuario en Kerberos, conseguir un ticket e ver a súa validez:
   root@kalib:~# kinit user1 #Autenticar como usuario **user1** e contrasinal **abc123.** conseguindo un ticket TGT. Ter en conta que o usuario *user1* xa existe na bbdd principal de kerberos e foi xerado de forma local en kalia mediante o comando *kadmin -l*
   user1@IES.LOCAL's Password:
   root@kalib:~# klist -l #Listar a validez do ticket TGT. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

Exemplo1. Máquina Virtual A: Kerberizar a Autenticación Local (pam-krb5)

i. Instalar módulo **libpam-krb5**:

root@kalia:~# apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
root@kalia:~# apt search libpam-krb5 || apt-cache search libpam-krb5 #Buscar calquera paquete que coincida co patrón de búsqueda libpam-krb5
root@kalia:~# apt -y install libpam-krb5 || apt-get -y install libpam-krb5 #Instalar o paquete libpam-krb5, é dicir, instalar o módulo PAM que permite aos usuarios locais autenticarse mediante o contrasinal de kerberos (e non a través de /etc/passwd) . Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

PAM y NSS | **Archivos de configuración utilizados por PAM y NSS** | **Bases de datos de usuarios y grupos** | **Módulos PAM (páxs 26,27)** | **YENDO MÁS ALLÁ NSS y bases de datos de sistema** (getent) | **Práctica Seguridade Informática - PAM**

ii. Configurar PAM con Kerberos:

Xa está feito! Unha vez instalado o paquete libpam-krb5 xa podemos autenticar mediante kerberos coas contas locais. Para iso, imos crear un usuario no sistema para validar con PAM auth (/etc/passwd) e imos crear un usuario co mesmo nome na bbdd principal de Kerberos con outro contrasinal. Veremos que podemos acceder co contrasinal de Kerberos á conta local.

root@kalia:~# pam-auth-update #Con esta comando podemos modificar/verificar os módulos de autenticación PAM activados.

root@kalia:~# apropos krb5 #Buscar en que páxinas do man e descripcións existen referencias ao nome dado: krb5
root@kalia:~# man pam_krb5 #Ver as páxinas do manual para pma_krb5

root@kalia:~# groupadd -g 1050 testing #Crear o grupo de nome *testing* co GID de valor 1050.
root@kalia:~# useradd -m -u 1050 -g 1050 -d /home/testuser -p $(mkpasswd -m sha-512 abc123.) -s /bin/bash testuser
#Crear o usuario *testuser* co comando *useradd*, onde:
-m → Copia na casa do usuario o que exista no cartafol /etc/skel
-u → Establece o valor do UID, neste caso 1050
-g → Establece o grupo principal, neste caso o valor GID 1050 que corresponde ao grupo *testing*
-d /home/testuser → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/testuser
-p $(mkpasswd -m sha-512 abc123.) → Pon *abc123.* como contrasinal cifrado (sha-512) para o login do usuario *testing*
-s /bin/bash → Establece como shell de traballo para o usuario a shell bash
testuser → Establece como nome de autenticación de usuario o nome testuser

root@kalia:~# getent passwd #Conseguir entradas de **Name Service Switch libraries**, neste caso conseguir os usuarios de passwd
root@kalia:~# getent passwd | grep testuser #Filtrar o comando anterior co patrón *testuser*, é dicir, amosa información de autenticación do usuario *testuser*
root@kalia:~# kadmin -l #Administrar de forma local kerberos

```
root@kalia:~# kadmin -l
kadmin> add testuser #Crear usuario testuser na bddd principal kerberos (base de datos de usuarios do KDC)
Max ticket life [1 day]:
Max renewable life [1 week]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
Policy [default]:
testuser@IES.LOCAL's Password: #Introducir o contrasinal que queremos que posúa na bbdd principal de Kerberos o usuario testuser. Por exemplo 123456
Verify password - testuser@IES.LOCAL's Password: #Introducir de novo o contrasinal anterior (123456)
kadmin> quit
root@kalia:~#
```

root@kalia:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kalia:~$ su - testuser #Acceder como usuario testuser pero **escribir o contrasinal kerberos de testuser: 123456**
testuser@kalia:~$ klist -l #Listar a validez do ticket TGT do usuario *testuser*. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

```
Name                 Cache name                     Expires

* testuser@IES.LOCAL   FILE:/tmp/krb5cc_1050_EbG96i   Dec 15 23:29:05 2020
```

Exemplo2. Máquina Virtual B: Kerberizar a Autenticación Local (pam-krb5)

i. Instalar módulo **libpam-krb5**:

root@kalib:~# apt update || apt-get update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
root@kalib:~# apt search libpam-krb5 || apt-cache search libpam-krb5 #Buscar calquera paquete que coincida co patrón de búsqueda libpam-krb5
root@kalib:~# apt -y install libpam-krb5 || apt-get -y install libpam-krb5 #Instalar o paquete libpam-krb5, é dicir, instalar o módulo PAM que permite aos usuarios locais autenticarse mediante o contrasinal de kerberos (e non a través de /etc/passwd) . Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.

PAM y NSS | **Archivos de configuración utilizados por PAM y NSS** | **Bases de datos de usuarios y grupos** | **Módulos PAM (páxs 26,27)** | **YENDO MÁS ALLÁ NSS y bases de datos de sistema** (getent) | **Práctica Seguridade Informática - PAM**

ii. Configurar PAM con Kerberos:

Xa está feito! Unha vez instalado o paquete libpam-krb5 xa podemos autenticar mediante kerberos coas contas locais. Para iso, imos crear un usuario no sistema para validar con PAM auth (/etc/passwd) e imos crear un usuario co mesmo nome na bbdd principal de Kerberos con outro contrasinal. Veremos que podemos acceder co contrasinal de Kerberos á conta local.

root@kalib:~# pam-auth-update #Con esta comando podemos modificar/verificar os módulos de autenticación PAM activados.

root@kalib:~# apropos krb5 #Buscar en que páxinas do man e descripcións existen referencias ao nome dado: krb5
root@kalib:~# man pam_krb5 #Ver as páxinas do manual para pma_krb5

root@kalib:~# groupadd -g 4000 untesting #Crear o grupo de nome *untesting* co GID de valor 4000.
root@kalib:~# useradd -m -u 4000 -g 4000 -d /home/testuser -p $(mkpasswd -m sha-512 abc123.) -s /bin/bash testuser
#Crear o usuario *testuser* co comando *useradd*, onde:
-m → Copia na casa do usuario o que exista no cartafol /etc/skel
-u → Establece o valor do UID, neste caso 4000
-g → Establece o grupo principal, neste caso o valor GID 4000 que corresponde ao grupo *untesting*
-d /home/testuser → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/testuser
-p $(mkpasswd -m sha-512 abc123.) → Pon *abc123.* como contrasinal cifrado (sha-512) para o login do usuario *testing*
-s /bin/bash → Establece como shell de traballo para o usuario a shell bash
testuser → Establece como nome de autenticación de usuario o nome testuser

root@kalib:~# getent passwd #Conseguir entradas de **Name Service Switch libraries**, neste caso conseguir os usuarios de passwd
root@kalib:~# getent passwd | grep testuser #Filtrar o comando anterior co patrón *testuser*, é dicir, amosa información de autenticación do usuario *testuser*

IMPORTANTE:
- Notar que o usuario debe existir localmente para que a autenticación poida realizarse, é dicir, o usuario existe e autentica pero neste caso na bbdd principal de Kerberos.
- Notar tamén que este usuario é local de kalib, polo que pode posuír outro uid, gid, grupos secundarios, contrasinal, casa de usuario... totalmente distintos ao usuario local de kalia.

root@kalib:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
kali@kalib:~$ su - testuser #Acceder como usuario testuser pero **escribir o contrasinal kerberos de testuser: 123456**

- klist -l #Listar a validez do ticket TGT do usuario *testuser*. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

```
Name                 Cache name                     Expires

* testuser@IES.LOCAL   FILE:/tmp/krb5cc_4000_a57zUd   Dec 16 14:34:02 2020
```

Exemplo3. Acceso SSH: Autenticación Kerberos. Contas de usuario locais kerberizadas e Servizo SSH non Kerberizado

Xa está feito! Unha vez instalado o paquete libpam-krb5 xa podemos autenticar mediante kerberos coas contas locais de forma local ou remota mediante conexións SSH. Para iso, imos comprobar que o usuario *testuser* pode autenticar con PAM auth (/etc/passwd) e tamén con Kerberos.

Dende a máquina virtual A. Servidor SSH:

i. **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 kalia.ies.local 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:~# /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.

ii. **Autenticación Kerberos sen servidor Kerberizado:**

root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal PAM (/etc/passwd). Se é a primeira vez 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.
root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal Kerberos. **Agora vemos que SI é posible a autenticación mediante Kerberos. Non fai falla configurar o servidor SSH para que o usuario testuser poida autenticar co seu contrasinal Kerberos.**

Dende a máquina virtual B. Cliente SSH:

i. **Autenticación Kerberos sen servidor Kerberizado:**

root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal PAM (/etc/passwd). Se é a primeira vez 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.
root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal Kerberos. **Agora vemos que SI é posible a autenticación mediante Kerberos. Non fai falla configurar o servidor SSH para que o usuario testuser poida autenticar co seu contrasinal Kerberos.**

Exemplo4. Kerberizar a Autenticación en Servizos (keytab)

Dende a máquina virtual A. Servidor Kerberos kalia.ies.local:

i. **Crear keytab para kalia.ies.local**:

(1) Crear principal host/kalia.ies.local

root@kalia:~# kadmin -l #Administrar de forma local kerberos

```
root@kalia:~# kadmin -l
kadmin> add -r host/kalia.ies.local@IES.LOCAL #Crear principal para o host kalia na bddd kerberos (KDC)
Max ticket life [1 day]:
Max renewable life [1 week]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
Policy [default]:
kadmin> quit
root@kalia:~#
```

(2) Exportar keytab do principal host/kalia.ies.local

root@kalia:~# kadmin -l #Administrar de forma local kerberos

```
root@kalia:~# kadmin -l
kadmin> ext host/kalia.ies.local@IES.LOCAL #Exportar a clave ao ficheiro /etc/krb5.keytab
kadmin> quit
root@kalia:~#
```

root@kalia:~# ls -l /etc/krb5.keytab #Listar de forma extendida o ficheiro /etc/krb5.keytab, o cal é o ficheiro por defecto onde se gardan as claves extraidas

root@kalia:~# ktutil -k /etc/krb5.keytab list #Listar o contido do ficheiro /etc/krb5.keytab

```
O arquivo de claves keytab pode ser xerado en calquera computadora que sexa cliente Kerberos e non ten porque vincularse a esta computadora onde foi xerado. Estes arquivos poden xerarse nunha computadora e copiarse para que poidan ser empregados por outras computadoras.
```

ii. **Crear keytab para kalib.ies.local**:

(1) Crear principal host/kalib.ies.local

root@kalia:~# kadmin -l #Administrar de forma local kerberos en kalia

```
root@kalia:~# kadmin -l
kadmin> add -r host/kalib.ies.local@IES.LOCAL #Crear principal para o host kalib na bddd principal kerberos (KDC)
Max ticket life [1 day]:
Max renewable life [1 week]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
Policy [default]:
kadmin> quit
root@kalia:~#
```

(2) Exportar keytab do principal host/kalib.ies.local

root@kalia:~# kadmin -l #Administrar de forma local kerberos

```
root@kalia:~# kadmin -l
kadmin> ext --keytab=/etc/krb5.keytab2 host/kalib.ies.local@IES.LOCAL #Exportar a clave ao ficheiro /etc/krb5.keytab2
kadmin> quit
root@kalia:~#
```

root@kalia:~# ls -l /etc/krb5.keytab2 #Listar de forma extendida o ficheiro /etc/krb5.keytab2, o cal é o ficheiro onde eliximos gardar a clave extraida do principal hostB/kaliB.ies.local@IES.LOCAL

root@kalia:~# ktutil -k /etc/krb5.keytab2 list #Listar o contido do ficheiro /etc/krb5.keytab2

root@kalia:~# cp -v /etc/krb5.keytab2 /home/kali/ #Copiar (modo verbose) o ficheiro /etc/krb5.keytab2 dentro do directorio /home/kali

root@kalia:~# chown kali /home/kali/krb5.keytab2 #Cambiar o usuario propietario do ficheiro /home/kali/krb5.keytab2 ao usuario kali

```
Agora debemos facer copia deste arquivo keytab de claves a kaliB para configurar os servizos que queiramos autenticar con Kerberos, no servidor Kerberos kalia.ies.local
```

Exemplo5. Kerberizar a Autenticación Remota SSH (/etc/ssh/sshd_config + keytab)

Dende a máquina virtual A. Servidor SSH:

i. **Autenticación Kerberos sen servidor Kerberizado:**

root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal PAM (/etc/passwd). Se é a primeira vez 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.

root@kalia:~# ssh -v testuser@kalia.ies.local #Comprobar se o servidor SSH está activo e podemos conectarnos a el co usuario testuser e o seu contrasinal Kerberos. **Agora vemos que SI é posible a autenticación mediante Kerberos. Non fai falla configurar o servidor SSH para que o usuario testuser poida autenticar co seu contrasinal Kerberos.**

ii. **Kerberizar Servidor SSH kalia.ies.local**:

(1) Modificar permisos keytab (/etc/krb5.keytab) do principal host/kalia.ies.local

root@kalia:~# chown sshd.ssh /etc/krb5.keytab #Cambiar usuario propietario sshd e grupo propietario ssh ao ficheiro /etc/krb5.keytab
root@kalia:~# chmod 640 /etc/krb5.keytab #Cambiar os permisos **ugo** do ficheiro krb5.keyta situado en /etc para establecer os permisos rw-r----- (lectura e escritura para o usuario propietario, soamente lectuar para o grupo propietario e ningún permiso para o resto do mundo)

(2) Modificar configuración Servidor SSH (/etc/ssh/sshd_config)

root@kalia:~# A=$(grep -n 'GSSAPIAuthentication' /etc/ssh/sshd_config | cut -d':' -f1 | xargs | awk '{print $NF}')
root@kalia:~# sed -i -e 's/GSSAPIAuthentication/##GSSAPIAuthentication/g' -e "${A}a\GSSAPIAuthentication yes" /etc/ssh/sshd_config #Cambiar a directiva a yes. Esta directiva permítenos realizar a autenticación mediante Kerberos
root@kalia:~# /etc/init.d/ssh reload #Recargar o ficheiro de configuración do servidor SSH para que se activen os cambios realizados

(3) Conseguir ticket TGT para o usuario testuser

root@kalia:~# kinit testuser #Autenticar en Kerberos como usuario **testuser** e contrasinal **123456** conseguindo un ticket TGT
root@kalia:~# klist -l #Listar a validez do ticket TGT. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

Dende a máquina virtual B. Cliente SSH:

i. **Copia a keytab /etc/krb5.keytab2 correspondente ao principal host/kalib.ies.local:**

root@kalib:~# scp kali@kalia.ies.local:krb5.keytab2 . #Contrasinal de acceso **abc123.** (Autenticación PAM).Estando situado no HostB, copiar de A a B (do servidor ao cliente) o arquivo /home/kali/krb5.keytab2, é dicir, copiar en B o ficheiro /home/kali/krb5.keytab2 existente no HostA, e copialo na ruta onde lanza o comando o usuario cliente que é o que simboliza o caracter '.' . Neste caso a copia realizarase no $HOME(~) do usuario root (/root)

root@kalib:~# cp -v krb5.keytab2 /etc/krb5.keytab #Copiar (modo verbose) o ficheiro krb5.keytab2 dentro do directorio /etc co nome krb5.keytab

ii. **Modificar permisos keytab /etc/krb5.keytab**:

(1) Modificar permisos keytab (/etc/krb5.keytab) do principal host/kalib.ies.local

root@kalib:~# chown sshd.ssh /etc/krb5.keytab #Cambiar usuario propietario sshd e grupo propietario ssh ao ficheiro /etc/krb5.keytab
root@kalib:~# chmod 640 /etc/krb5.keytab #Cambiar os permisos **ugo** do ficheiro krb5.keyta situado en /etc para establecer os permisos rw-r----- (lectura e escritura para o usuario propietario, soamente lectuar para o grupo propietario e ningún permiso para o resto do mundo)

(2) Modificar configuración Cliente SSH (/etc/ssh/ssh_config)

root@kalib:~# A=$(grep -n 'GSSAPIAuthentication' /etc/ssh/sshd_config | cut -d':' -f1 | xargs | awk '{print $NF}')
root@kalib:~# sed -i -e 's/GSSAPIAuthentication/##GSSAPIAuthentication/g' -e "${A}a\GSSAPIAuthentication yes" /etc/ssh/sshd_config #Cambiar a directiva a yes. Esta directiva permítenos realizar a autenticación mediante Kerberos
root@kalib:~# A=$(grep -n 'GSSAPIDelegateCredentials' /etc/ssh/ssh_config | cut -d':' -f1 | xargs | awk '{print $NF}')
root@kalib:~# sed -i -e 's/GSSAPIDelegateCredentials/##GSSAPIDelegateCredentials/g' -e "${A}a\GSSAPIDelegateCredentials yes" /etc/ssh/ssh_config #Cambiar a directiva a yes. Esta directiva permítenos realizar a autenticación mediante Kerberos

(3) Conseguir ticket TGT para o usuario testuser

root@kalib:~# kinit testuser #Autenticar en Kerberos como usuario **testuser** e contrasinal **123456** conseguindo un ticket TGT
root@kalib:~# klist -l #Listar a validez do ticket TGT. Este ticket é válido para acceder a calquera servizo de rede que empregue Kerberos para autenticación.

iii. **Conexión mediante Autenticación Kerberos + Servizo SSH Kerberizado**

root@kalib:~# ssh -K -v testuser@kalia.ies.local #Acceso sen contrasinal mediante autenticación servidor kerberizado, é dicir, o usuario testuser accede directamente sen escribir ningún contrasinal. A opción -K permite a autenticación Kerberos (basada en GSSAPI) e o reenvío de credenciais ao servidor.

testuser@kalia:~$

Ricardo Feijoo Costa

Apéndice. URLs Cheat Sheet

Cheat Sheet ISC DHCP Server

Cheat Sheet DNS GNU/Linux

Cheat Sheet Apache2 GNU/Linux

Ricardo Feijoo Costa

Comandos apt, dpkg

Práctica apt

$ apt #Amosar o uso do comando apt

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

apt update #Actualizar lista de paquetes dispoñibles a instalar dos repositorios

apt search refcard #Buscar nas descricións dos paquetes o patrón refcard

apt show debian-refcard #Amosar detalles do paquete debian-refcard

apt install debian-refcard #Instalar o paquete debian-refcard

apt reinstall debian-refcard #Reinstalar o paquete debian-refcard

apt remove debian-refcard #Eliminar o paquete debian-refcard

apt install debian-refcard #Instalar o paquete debian-refcard

apt purge debian-refcard #Eliminar e purgar o paquete debian-refcard. É equivalente á opción remove pero ademais elimina calquera arquivo de configuración do paquete

apt policy debian-refcard #Amosa o posible paquete candidato de debian-refcard a instalar e, dado o caso, que versión do paquete debian-refcard está instalado.

apt autoremove #Eliminar automaticamente tódolos paquetes que foron instalados para satisfacer dependencias para outros paquetes e xa non son necesarios, ben porque as dependencias cambiaron ou os paquetes que os necesitaban xa foron borrados

apt -y upgrade #Actualizar o sistema instalando/actualizando paquetes respostando **yes** a tódalas posibles preguntas durante a actualización

apt -y full-upgrade #Actualizar o sistema eliminando/instalando/actualizando paquetes respostando **yes** a tódalas posibles preguntas durante a actualización

exit #Pechar a consola de comandos do usuarío actual, neste caso do usuario root

Práctica apt-get

$ apt-get #Amosar o uso do comando apt-get

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

apt-get update #Actualizar lista de paquetes a instalar dos repositorios

apt-get -y upgrade #Actualizar o sistema (todos os ficheiros actualmente instalados) respostando **yes** a tódalas posibles preguntas durante a actualización

apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

apt-get -y remove debian-refcard #Eliminar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

apt-get -y purge debian-refcard #Eliminar e purgar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación. É equivalente á opción remove pero ademais elimina calquera arquivo de configuración do paquete

apt-get clean #Limpar os ficheiros descargados no repositorio local, é dicir, elimina todos os ficheiros existentes nas rutas /var/cache/apt/archives/ e /var/cache/apt/archives/partial/

apt-get autoclean #Similar ao comando anterior, pero soamente elimina os arquivos dos paquetes que xa non se poden descargar e levan tempo sen usar

apt-get autoremove #Eliminar automaticamente tódolos paquetes que foron instalados para satisfacer dependencias para outros paquetes e xa non son necesarios, ben porque as dependencias cambiaron ou os paquetes que os necesitaban xa foron borrados

exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$ mkdir /tmp/paquetes && cd /tmp/paquetes #Crear o cartafol /tmp/paquetes e no caso que o comando teña éxito, é dicir, sexa executado sen erros, farase o segundo comando o cal accede ao directorio /tmp/paquetes

$ apt-get download debian-refcard #Descargar o ficheiro deb do paquete buscado debian-refcard na ruta actual. Descargará soamente o paquete e non as súas dependencias.

Práctica apt-cache

$ apt-cache #Amosar o uso do comando apt-cache. Este comando non precisa ser executado con permisos de root

$ apt-cache policy debian-refcard #Amosa o posible paquete candidato de debian-refcard a instalar e, dado o caso, que versión do paquete debian-refcard está instalado.

$ apt-cache search refcard #Buscar nas descricións dos paquetes o patrón refcard

$ apt-cache show debian-refcard #Amosar detalles do paquete debian-refcard

$ apt-cache depends debian-refcard #Amosar información de dependencias do paquete debian-refcard

$ apt-cache depends ntp #Amosar información de dependencias do paquete ntp

$ apt-cache rdepends debian-refcard #Amosar información de dependencias inversas do paquete debian-refcard

$ apt-cache rdepends ntp #Amosar información de dependencias inversas do paquete ntp

$ apt-cache depends debian-refcard ntp #Amosar información de dependencias dos paquetes debian-refcard e ntp

$ apt-cache rdepends debian-refcard ntp #Amosar información de dependencias inversas dos paquetes debian-refcard e ntp

$ apt-get update #Actualizar lista de paquetes a instalar dos repositorios. **Amosa erro porque este comando precisa ser executado con permisos de root.**

$ apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación. **Amosa erro porque este comando precisa ser executado con permisos de root.**

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

apt-get -y install debian-refcard terminator #Instalar os paquetes debian-refcard e terminator respostando **yes** a tódalas posibles preguntas durante a instalación

exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

Práctica dpkg

$ dpkg --help #Amosar a axuda do comando dpkg

$ dpkg -l debian-refcard #Listar información sobre o paquete debian-refcard

$ dpkg -L debian-refcard #Listar os ficheiros pertencentes ao paquete debian-refcard

$ dpkg -S /usr/share/doc/debian-refcard/refcard-es-a4.pdf.gz #Atopar o/s paquete/s propietario/s do/s ficheiro/s buscado/s.

$ dpkg -l debian-refcard terminator #Listar información sobre os paquetes debian-refcard e terminator

```
$ dpkg -L debian-refcard terminator #Listar os ficheiros pertencentes aos paquetes debian-refcard e terminator

$ dpkg -S /usr/share/doc/debian-refcard/refcard-es-a4.pdf.gz /usr/bin/terminator #Atopar o/s paquete/s propietario/s do/s ficheiro/s buscado/s.

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

    # dpkg -r debian-refcard #Eliminar o paquete debian-refcard

    # dpkg -l debian-refcard #Listar información sobre o paquete debian-refcard

    # cd /tmp/paquetes #Cambiar ao directorio /tmp/paguetes

    # dpkg -i $(ls debian-refcard*) #Instalar o paquete descargado debian-refcard.

    # dpkg -l debian-refcard #Listar información sobre o paquete debian-refcard

    # dpkg -P debian-refcard #Eliminar e purgar o paquete debian-refcard. É equivalente á opción remove pero ademais elimina calquera arquivo de configuración do paquete

    # dpkg -l debian-refcard #Listar información sobre o paquete debian-refcard

    # exit #Pechar a consola de comandos do usuarío actual, neste caso do usuario root

$
```

Ricardo Feijoo Costa