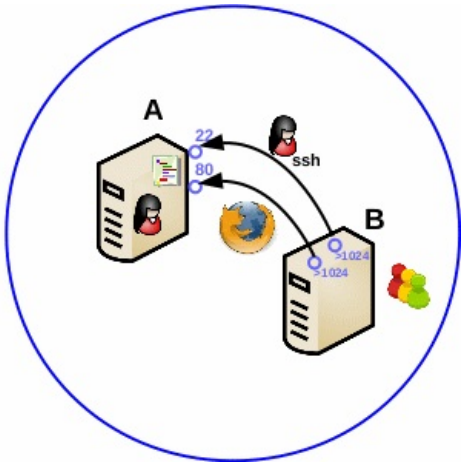
Conexión Remota mediante SSH
Cambios en ssh_config

ESCENARIO

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

Máquina virtual A:	**Máquina virtual B:**
Rede Interna	Rede Interna
Servidor SSH: openssh-server	Cliente SSH: openssh-client (ssh)
Servidor Web: Apache (apache2)	Cliente Web: Navegador (firefox)
ISO: Kali Live amd64	ISO: Kali Live amd64
IP/MS: 192.168.120.100/24	IP/MS: 192.168.120.101/24

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



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

NOTAS:

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

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

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

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

root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
root@kaliA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)
root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
root@kaliA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled
root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
root@kaliA:~# ssh -v kali@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario kali e o seu contrasinal. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.
root@kaliA:~# exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

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:
   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**.

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 #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 #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@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 **root**.

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 * #As 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* establecéndose 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 * #As 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 *)* e 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 *)* e 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 * #As 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.

Ricardo Feijoo Costa