TALLER BRS
PRÁCTICA SSH 2FA + Cifrado Asimétrico

Apelidos	Nome

ESCENARIO

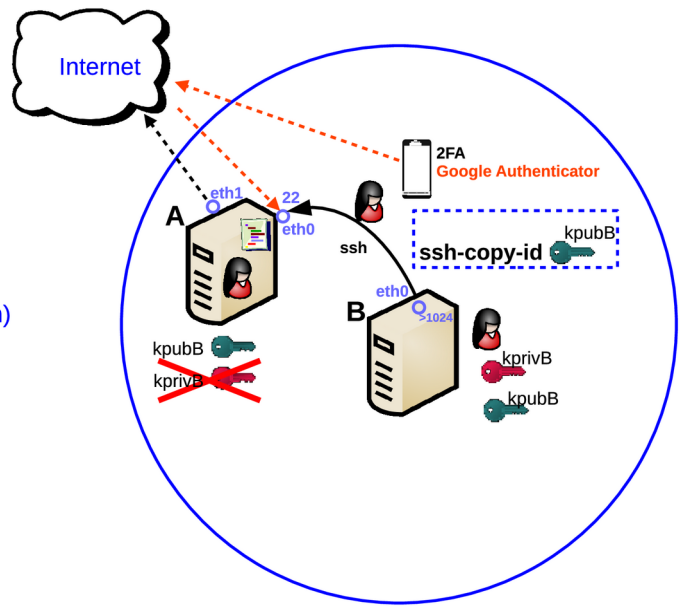
Máquinas virtuais:
RAM ≥ 2048MB CPU ≥ 2 PAE/NX habilitado
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina virtual A:
Rede: Modo NAT (10.0.2.0/24)
  eth0→IP/MS: 10.0.2.15/24
Rede Interna: 10.0.0.0/8
  eth1→IP/MS: 10.10.10.10/8
Servidor SSH: openssh-server
ISO: Kali Live amd64
2FA: libpam-google-authenticator
id_rsa.pub=kpubB

Máquina virtual B:
Rede Interna: 10.0.0.0/8
  eth0→IP/MS: 10.10.10.10/8
Cliente SSH: openssh-client (ssh)
ISO: Kali Live amd64

id_rsa.pub = kpubB
id_rsa = kprivB

Móbil:
Google Authenticator



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.

Material necesario	Práctica: SSH 2FA + Cifrado Asimétrico
■ Host alumnado ■ Máquinas virtuais GNU/Linux Kali ■ [1] Criptografía ■ [2] Criptografía - SSH ■ [3] 2FA – Google Authenticator ■ [4] ISO descarga GNU/Linux Kali ■ [5] PAM ■ [6] Emulador Android for PC & MAC ■ [7] Combining 2FA and Public Key Authentication for a better Linux SSH security	Host alumnado: a) Máquina virtual A GNU/Linux Kali amd64 [4]: ■ Crear seguindo especificacións do escenario. ■ Arrancar ■ Configurar a rede según o escenario ■ Arrancar servidor SSH Host alumnado: b) Máquina virtual B GNU/Linux Kali amd64 [4]: ■ Crear seguindo especificacións do escenario. ■ Arrancar ■ Configurar a rede según o escenario ■ Conectar ao servidor SSH mediante cifrado asimétrico Host alumnado: c) Máquina virtual A GNU/Linux Kali amd64 [4] ■ Configurar 2FA no servidor SSH Emulador[6]/Móbil: Capturar código QR con Google Authenticator [3] Host alumnado: d) Máquina virtual A GNU/Linux Kali amd64 [4] ■ Configurar 2FA no servidor SSH con cifrado asimétrico Host alumnado: e) Máquina virtual B GNU/Linux Kali amd64 [4] ■ Comprobar que é posible o acceso ao servidor SSH mediante 2FA – Google Authenticator [3]: Introducir contrasinal do usuario Emulador[6]/Móbil: ■ Comprobar que é posible o acceso ao servidor SSH mediante 2FA – Google Authenticator [3]: Introducir código ofrecido por Google Authenticator

Procedemento:

(1) Host alumnado. Máquina virtual GNU/Linux Kali:

(a) Crear e arrancar unha máquina virtual no equipo do alumnado coas seguintes características (ver escenario):

i. RAM ≥ 4096MB
ii. CPU ≥ 2
iii. PAE/NX habilitado
iv. Rede: 2 tarxetas de rede,
   - eth0 → NAT
   - eth1 → Rede Interna
v. ISO: Kali Live amd64 [4]
vi. Nome: Servidor-SSH-2FA+Cifrado-Asimétrico

(b) O xestor de redes NetworkManager está habilitado. Por defecto, está xerada unha conexión da interface eth0 solicitando a configuración de rede mediante DHCP, e como temos a tarxeta eth0 en modo NAT deberiamos obter a IP 10.0.2.15 e ter conexión a Internet. Así, executar nunha consola:

```
$ setxkbmap es #Configurar teclado en español

$ ip addr show #Amosar información sobre as NIC existentes no sistema, é dicir, verificar a configuración de rede  para as NIC: lo, eth0 e eth1

$ ip addr show eth0 #Amosar información sobre a NIC eth0. Verificar que a configuración de rede para a NIC eth0 é a seguinte: IP=10.0.2.15, MS=255.255.255.0

$ ip route #Ver a táboa de rutas do sistema.Verificar que GW=10.0.2.2

$ cat /etc/resolv.conf #Ver o contido do ficheiro /etc/resolv.conf, o cal contén a configuración os servidores DNS a empregar para a resolución de nomes. Comprobar que as directivas nameserver coinciden cos DNS1 e DNS2 da aula taller.
```

(c) Imos xerar unha configuración de rede manual. Así, executar na consola anterior:

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

  # /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.

  # /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.

  # ip addr add 10.10.10.10/8 dev eth1 #Configurar a tarxeta de rede eth1, coa IP: 10.10.10.10 e máscara de subrede: 255.0.0.0

  # ip addr show eth1 #Amosar información sobre a NIC eth1. Verificar a configuración de rede  para a NIC eth1

  # /etc/init.d/ssh start || systemctl start ssh #Iniciamos o servidor SSH xa que non está arrancado. Se estiverá arrancado e se modificamos o ficheiro de configuración poderíamos executar o seguinte comando para recargar a configuración do servidor SSH:

     /etc/init.d/ssh reload || systemctl reload ssh

  # exit #Saír da shell

$
```

(2) Hosts alumnado. Máquina virtual GNU/Linux Kali:

(a) Crear e arrancar unha máquina virtual no equipo do alumnado coas seguintes características (ver escenario):

i. RAM ≥ 4096MB
ii. CPU ≥ 2
iii. PAE/NX habilitado
iv. Rede: 1 tarxeta de rede(eth0) en modo Rede Interna
v. ISO: Kali Live amd64 [4]
vi. Nome: Cliente-SSH-2FA+Cifrado-Asimétrico

(b) O xestor de redes NetworkManager está habilitado. Por defecto, está xerada unha conexión da interface eth0 solicitando a configuración de rede mediante DHCP, e como temos a tarxeta eth0 en modo Rede Interna debemos configurala polo que imos xerar unha configuración de rede manual. Así, executar nunha consola:

```
$ setxkbmap es #Configurar teclado en español

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

  # /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.

  # /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.

  # ip addr add 10.10.10.11/8 dev eth0 #Configurar a tarxeta de rede eth0, coa IP: 10.10.10.11 e máscara de subrede: 255.0.0.0

  # ip addr show eth0 #Amosar información sobre a NIC eth0. Verificar a configuración de rede para a NIC eth0

  # exit #Saír da shell

$
```

(c) Configurar cifrado asimétrico:

```
$ 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.

```
$ 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**

```
$ ssh-copy-id -i .ssh/id_rsa.pub kali@10.10.10.10 #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 kali
- Agora a conexión sen contrasinal será posible para o usuario kali, con todos os permisos deste usuario, na máquina Servidor SSH (10.10.10.10).

```
$ ssh -v kali@10.10.10.10 #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende a máquina B 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.

    $ exit #Saír da consola remota ssh a que acabamos de acceder mediante cifrado asimétrico, para v-oltar á consola local de kali na máquina B.

$ ssh -v -i .ssh/id_rsa kali@10.10.10.10 #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende a máquina B sen contrasinal, a través de cifrado asimétrico. Agora indícase onde se pode atopar a clave privada para a autenticación mediante a opción -i

    $ exit #Saír da consola remota ssh a que acabamos de acceder mediante cifrado asimétrico, para voltar á consola local de kali na máquina B.
```

(3) Host alumnado. Máquina virtual GNU/Linux Kali: Servidor-SSH-2FA+Cifrado-Asimétrico

(a) Configurar 2FA no servidor SSH [2]. Executar na anterior consola:

```
#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 -y install libpam-google-authenticator \

|| apt-get -y install libpam-google-authenticator #Instalar o paquete de nome libpam-google-authenticator. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete. libpam-google-authenticator é un paquete fundamental para implementar a autenticación de dous factores (2FA) en sistemas GNU/Linux, utilizando a popular aplicación Google Authenticator. Esta ferramenta engade un nivel extra de seguridade ás túas sesións de inicio, requirindo non só a túa contrasinal, senón tamén un código de verificación de un só uso (OTP) xerado por unha aplicación no teu teléfono móbil.

# exit #Saír da shell

$ google-authenticator # Respostando y ás cuestións que nos ofrece, o comando, estamos a asegurar o comportamento do 2FA, gardando a configuración escollida no ficheiro $HOME/.google-authenticator

Do you want authentication tokens to be time-based (y/n) y

Warning: pasting the following URL into your browser exposes the OTP secret to Google:

https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/kali@kali%3Fsecret%3DD6LWY2JK75MJQHXQXVJ3KARRLQ%26issuer%3Dkali
```

```
Your new secret key is: D6LWY2JK75MJQHXQXVJ3KARRLQ
```

(b) Agora debemos escanear o código QR no móbil coa aplicación Google-Authenticator [3] e introducir o código que nos ofrece Google-Authenticator (cada 30 segundos):

```
Enter code from app (-1 to skip): 878576
Code confirmed
Your emergency scratch codes are:
    83283719
    62825457
    87144405
    88051009
    49460991
Do you want me to update your "/home/kali/.google_authenticator" file? (y/n) y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, a new token is generated every 30 seconds by the mobile app.
In order to compensate for possible time-skew between the client and the server,
we allow an extra token before and after the current time. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) y

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n) y
```

```
$ cat .google_authenticator
D6LWY2JK75MJQHXQXVJ3KARRLQ
" RATE_LIMIT 3 30
" WINDOW_SIZE 17
" DISALLOW_REUSE
" TOTP_AUTH
83283719
62825457
87144405
88051009
49460991
```

(c) Configurar 2FA no servidor SSH [2]. Executar na anterior consola:

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

`# sed -i 's/KbdInteractiveAuthentication no/KbdInteractiveAuthentication yes/' \ /etc/ssh/sshd_config` #Configura a yes a directiva KbdInteractiveAuthentication, a cal permite:

- **Autenticación multifactorial**: Solicitar ao usuario un segundo factor de autenticación, como un código de un token de seguridade ou unha aplicación de autenticación.
- **Autenticación basada en preguntas e respostas**: Presentar ao usuario unha serie de preguntas que soamente el debería coñecer.
- **Integración con sistemas de autenticación externos**: Utilizar módulos PAM (Pluggable Authentication Modules)[5] para autenticar ao usuario contra outros sistemas.

Esta directiva é unha versión moderna da directiva **ChallengeResponseAuthentication** (que tamén debería estar posta a yes)

`# echo -e '\n# Cifrado_Asimétrico+2FA\nAuthenticationMethods publickey,keyboard-interactive' >> /etc/ssh/sshd_config` #Engadimos a configuración necesaria para que o servizo SSH esté obrigado a empregar ambos: cifrado asimétrico e 2FA

`# sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/' >> /etc/ssh/sshd_config` #Modificamos o valor da directiva PasswordAuthentication a no para deshabilitar o acceso por contrasinal.

`# echo -e '\n# 2FA Google Authenticator\nauth required pam_google_authenticator.so' \ >> /etc/pam.d/sshd` #Engadimos a configuración necesaria para que o servizo SSH esté obrigado a exigir a configuración 2FA, onde a liña engadida `auth required pam_google_authenticator.so` é a parte clave da configuración, onde:

- auth: Indica que esta liña aplícase a fase de autenticación.
- required: Significa que o módulo pam_google_authenticator.so é obrigatorio para a autenticación. Se falla, o usuario non poderá iniciar sesión.
- pam_google_authenticator.so: Este é o módulo PAM que se cargará para realizar a autenticación de dous factores utilizando Google Authenticator.

`# sed -i -e 's/@include common-auth/#@include common-auth/' /etc/pam.d/sshd` #Comentar esa liña en /etc/pam.d/sshd para non incluír o arquivo common-auth que soe conter configuracións de autenticación comúns que son utilizadas por múltiples servizos. Ao descomentalo estás agregando esas configuracións á configuración de SSH. Isto pode incluir módulos para autenticación contra bases de datos, LDAP, ou outras fontes de autenticación.

`# sed -i -e 's/account required pam_access.so/#account required pam_access.so/' /etc/pam.d/sshd` #Comentar esa liña en /etc/pam.d/sshd para que este módulo que permite controlar o acceso aos servizos basados en regras definidas no ficheiro /etc/security/access.conf non se teña en contas. Así, pódese permitir ou denegar o acceso a certos usuarios ou grupos en función da súa hora de conexión, terminal e dirección IP.

`# sed -i -e 's/@include common-password/#@include common-password/' /etc/pam.d/sshd` #Comentar esa liña en /etc/pam.d/sshd para non incluír o arquivo que soe conter configuracións relacionadas coa xestión de contrasinais, cambio de contrasinais, etc.

`# /etc/init.d/ssh restart || systemctl restart ssh` #Reiniciar o servidor SSH

(4) Host alumnado. Máquina virtual GNU/Linux Kali: Cliente-SSH-2FA+Cifrado-Asimétrico

(a) Conectar ao servidor SSH. Así, executar na anterior consola:

`$ ssh kali@10.10.10.10` #Comprobamos o acceso ao servidor SSH co usuario kali dende esta máquina cliente: 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.

```
The authenticity of host '10.10.10.10 (::1)' can't be established.
ED25519 key fingerprint is SHA256:wNbvnFlVdaAXsPRXVCHeRikyyOPysv0mAJVIEizm6oo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
Warning: Permanently added '10.10.10.10' (ED25519) to the list of known hosts.
(kali@10.10.10.10) Verification code: (Aquí debemos introducir o código de Google Authenticator)
```

```
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 25 22:37:31 2024 from 10.10.10.11
┌──(kali㉿kali)-[~]
└─$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:bb:29:7b brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.10/8 scope global eth1
    valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ exit
Connection to 10.10.10.10 closed.
$
```