UABCSIRT
Universitat Autònoma de Barcelona

Configuració segura de servidor SSH

20 de març 2017

Contents

A. Instal·lació i Configuració servidor SSH

L'objecte de la present guia és proporcionar als administradors de sistemes una guia que permeti definir una configuració bàsica en termes de seguretat per al servei SSH. Com veurem a la present guia SSH no nomes ens servirà per connectar-nos a la nostre màquina remota de manera segura, sinó que ens permetrà fer transferència d'arxius mitjançant SCP o sFTP.

1. Instal·lació

Per poder connectar-nos al nostre servidor utilitzarem OpenSSH, per instal·lar-ho haurem de fer les següents comandes.

```
sudo apt-get update
sudo apt-get install openssh-server
```

Una vegada instal·lat el servidor ssh editarem sshd_config per configurar els diferents paràmetres de seguretat que s'han de tindre en compte. Aquest es un exemple de configuració però haurem d'adaptar alguns paràmetres al nostre cas, com IPs, usuaris, etc.

1.1. Configuració

```
sudo gedit /etc/ssh/sshd_config

# Una manera de descartar la majoria dels bots que van rastrejant els serveis SSH
# oberts a les xarxes és fent el canvi de port per on ens connectarem al servei:
Port 1234
# A més si sabem des de quina IP ens connectarem al nostre servidor (tant interna
# com externa) només ens caldrà especificar-ho amb les següents comandes:
ListenAddress 192.168.1.50
ListenAddress 80.81.82.83
# El següent paràmetre que haurem d'introduir ens obligarà a utilitzar el protocol 2
#de SSH, i una criptografia més robusta. Després s'especificaran les rutes on es
#guardaran les keys.
Protocol 2
# FCS_SSH_EXT.1.6
Ciphers aes256-ctr,aes128-ctr
# FCS_SSH_EXT.1.8
```

```
MACs hmac-sha1,hmac-sha2-256,hmac-sha2-512

# HostKeys for protocol version 2

HostKey /etc/ssh/ssh_host_rsa_key

#HostKey /etc/ssh/ssh_host_dsa_key

HostKey /etc/ssh/ssh_host_ecdsa_key

HostKey /etc/ssh/ssh_host_ed25519_key

#Privilege Separation is turned on for security

UsePrivilegeSeparation yes

# Ens servirà per veure les autenticacions fallies i poder actuar sobre elles,

# s'explicarà més en detall a la instal·lació de fail2ban.

SyslogFacility AUTH

LogLevel INFO

# Authentication:

# En la part d'autenticació tenim el PermitRootLogin, molt recomanable deixar-ho
#desactivat, ja que per atacs de força bruta quan es detecta el servei ssh les
#comprovacions es realitzen amb aquest usuari, per tant ens crearem un usuari personal
#per accedir al servidor i quan ho necessitem ens donarem privilegis de root
#mitjançant su o sudo. Recomanable no utilitzar noms d'usuari admin, manager,
#sistemes, etc.

PermitRootLogin no

#La patalla de login es mantindrà oberta durant 30 segons.

LoginGraceTime 30

StrictModes yes

# Don't read the user's ~/.rhosts and ~/.shosts files

IgnoreRhosts yes

# For this to work you will also need host keys in /etc/ssh_known_hosts

RhostsRSAAuthentication no

# similar for protocol version 2

HostbasedAuthentication no

# Limitarem el número d'intents per encertar les credencials, encara que una vegada
#el servidor ens tanca la sessió podríem tornar a establir novament la connexió i
#tornar-ho a provar per tant recorreríem a el software fail2ban per regular-ho.

MaxAuthTries 2

# Indicarem el numero màxim d'usuaris que poden estar connectats al servidor.

MaxSessions 3
```

```
UsePAM yes
IgnoreRhosts yes
PermitEmptyPasswords no
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*
# override default of no subsystems
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

1.2. Instal·lació Fail2ban

En el cas anterior s'ha posat un exemple de configuració mitjançant contrasenya, per afegir més seguretat per aquest sistema es recomana la utilització de fail2ban, un programari que ens permetrà bloquejar atacs de força bruta bloquejant les IPs (iptables) que facis una sèrie d'autenticacions fallides en el nostre sistema basant-se en els logs que s'han habilitat a l'arxiu de configuració anterior.

```
sudo apt-get install fail2ban

sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

A l'arxiu jail.local enganxarem la següent configuració, amb el que s'estableix 3 com a número màxim d'autenticacions fallides abans de crear una regla de bloqueig per la IP especifica i augmentem el temps de bloqueig a 3600. També tenim altres paràmetres com ignoreip on podem especificar les IPs que no rebran denegació del servidor.

```
# "bantime" is the number of seconds that a host is banned.
bantime  = 3600

# "maxretry" is the number of failures before a host get banned.
maxretry = 3
```

1.3. Autenticació mitjançant pubkeys

Aquest mètode d'autenticació es complementari a la utilització de contrasenyes però si podem fer us únicament de validació amb claus publico/privades garantirem una major seguretat en les connexions amb el nostre servidor, ja que un atacant ha de disposar de la nostra clau privada així com el passphrase. En cas de que optem per una validació mitjançant contrasenya ens podem descarregar el document 'Politica de credencials UAB' per assegurar-nos de tindre una paraula de pas robusta.

1.3.1. Configuració Client

El primer pas per autenticar-nos amb una clau publico/privada és generar aquestes, per fer-ho el paquet openssh-clien proporciona la utilitat ssh-keygen amb la que generarem una clau amb encriptació RSA de 4096 bits encara que també es poden generar altres clau com DSA, ECDSA, etc.

Una vegada executada la comanda ens demanarà si volem guardar les claus en el directori per defecte o en un altre directori, com a tercer i quart punt normalment si es per tasques com realització de backups automàtics dins de la intranet s'acostuma a deixar en blanc ja que per cada tasca s'haurà d'introduir la paraula de pas, encara que si ens connectem des de internet es recomanable posar-la ja que a part de la contrasenya un atacant també hauria d'aconseguir el certificat públic per poder-se connectar al servidor.

```
ssh-keygen -t rsa -b 4096
01 Generating public/private dsa key pair.
02 Enter file in which to save the key (/home/usuari/.ssh/id_rsa):
03 Enter passphrase (empty for no passphrase):
04 Enter same passphrase again:
05 Your identification has been saved in /home/usuari/.ssh/id_rsa.
06 Your public key has been saved in /home/usuari/.ssh/id_rsa.pub.
07 The key fingerprint is:
08 e7:0e:2e:d6:aa:90:6e:9b:ac:ad:7f:6f:1d:23:50:28 usuari@ua
```

En el cas que s'opti per la utilització de passphrase es pot fer us de ssh-agent i ssh-add per a no haver de repetir la la passphrase en cada operació.

```
[usuari@client ~]$ ssh-agent /bin/bash
[usuari@client ~]$ ssh-add
Enter passphrase for /home/usuari/.ssh/id_dsa:
```

1.3.2. Configuració Servidor

Ara que tenim generades les claus corresponents, es necessari copiar la clau publica generada (id_rsa.pup) en el nostre servidor SSH al que ens volem connectar. Per tant guardarem aquesta clau publica dins el directori ~/.ssh/authorized_keys del usuari corresponent, en aquesta carpeta podem posar tantes claus publiques com desitgem. En el cas de que no tinguem creada la carpeta en el compte del usuari corresponent procedirem a crear-la amb permisos per aquell mateix usuari i la carpeta authorized_keys amb permisos més restrictius.

```
usuari@servidor $> pwd
/home/usuari
usuari@servidor $> mkdir .ssh; chmod 700 .ssh
usuari@servidor $> cd .ssh
usuari@servidor $> touch authorized_keys; chmod 600 authorized_keys
```

Finalment en copiarem la clau publica al servidor.

```
usuari@cliente $> cd ~/.ssh
usuari@cliente $> cat id_dsa.pub | ssh usuari@servidor "cat - >> ~/.ssh/authorized_keys"
```

UABCSIRT
Universitat Autònoma de Barcelona

INNOVATION MAKERS