Trabajo con claves ssh

Este apartado es **opcional** y recomendado para aquellos participantes del curso que no estén habituados a trabajar con pares de claves RSA en conexiones ssh. Ssh es una maravilla de aplicación que permite acceder a sistemas tipo UNIX de forma remota y segura, pudiendo utilizar diferentes mecanismos para autenticar el usuario siendo la contraseña y el par de claves pública/privada los dos mecanismos más usados. A continuación explicaremos las diferentes formas de utilizar acceso con pares de claves pública/privada.

Acceso por ssh con contraseña

La forma más simple de acceder por ssh es utilizando la contraseña del usuario en el equipo remoto:

```
usuario@cliente:~$ ssh usuario@10.0.1.8
The authenticity of host '10.0.1.8 (10.0.1.8)' can't be established.
ECDSA key fingerprint is 53:b8:8c:c7:52:32:39:ca:7c:79:92:d3:48:92:5b:da.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.8' (ECDSA) to the list of known hosts.
usuario@10.0.1.8's password:
Linux wheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.35-2 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
usuario@servidor:~$
```

Vamos a explicar los pasos que se han dado en la conexión anterior:

- No se reconoce la clave pública del servidor ssh remoto (cuya huella es 53:b8:8c:c7:52:32:39:ca:7c:79:92:d3:48:92:5b:da), pero se ofrece al usuario la posibilidad de aceptarlo como un servidor válido bajo su propia responsabilidad.
- Si el usuario acepta la clave pública remota se añadirá la clave pública del servidor ssh al fichero ~/.ssh/known_hosts de la cuenta del usuario en el equipo cliente.
- Posteriormente se solicita la contraseña del usuario en el servidor y si es correcta se inicia la sesión remota.

Este método es sencillo y seguro pero no es válido para procesos no interactivos o para usuarios sin contraseña, es decir, siempre es necesario que haya una persona esperando la solicitud de la contraseña para escribirla y es imprescindible que el usuario del equipo remoto tenga definida una contraseña, cosa que no ocurre siempre sobre todo con usuarios vinculados a servicios o similares.

Acceso por ssh con clave pública RSA sin frase de paso

Una forma muy cómoda de evitar las limitaciones del caso anterior es utilizar un par de claves pública/privada cifradas con el algoritmo RSA (también es posible utilizar DSA), por lo que en primer lugar generamos el par de claves RSA en el equipo cliente, dejamos los valores por defecto para la ubicación y nombres de las claves y no definimos frase de paso:

```
usuario@cliente:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/usuario/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/usuario/.ssh/id_rsa.
Your public key has been saved in /home/usuario/.ssh/id_rsa.pub.
The key fingerprint is:
46:bd:c0:2b:66:76:10:91:46:b3:b2:6d:35:1e:ad:8c usuario@cliente
The key's randomart image is:
+--[ RSA 2048]----+
|     .=o         |
|      o= o       |
|    ..o B o      |
|     + B * .     |
|    . E S .      |
|     = +         |
|                 |
|                 |
|                 |
+-----------------+
```

Si hacemos ahora un listado de los ficheros del directorio ~/.ssh, aparecerán tres ficheros:

```
usuario@cliente:~$ ls -l ~/.ssh/
total 12
-rw------- 1 usuario usuario 1675 feb  7 20:28 id_rsa
-rw-r--r-- 1 usuario usuario  397 feb  7 20:28 id_rsa.pub
-rw-r--r-- 1 usuario usuario  222 feb  7 20:08 known_hosts
```

id_rsa es la clave privada del usuario adecuadamente protegida (permisos 0600) e id_rsa.pub es la clave pública del usuario que debemos copiar en el equipo servidor, además aparece el fichero known_hosts en el que se ha almacenado la clave pública del servidor que aceptamos anteriormente. Vamos a proceder a copiar la clave pública del usuario en el equipo servidor utilizando ssh-copy-id, aunque se podría hacer de diversas maneras:

```
usuario@cliente:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub usuario@10.0.1.8
usuario@10.0.1.8's password:
Now try logging into the machine, with "ssh 'usuario@10.0.1.8'", and check in:

~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Los mensajes que aparecen nos informan que se ha copiado la clave pública al servidor y que debemos poder acceder ahora sin necesidad de utilizar la contraseña:

```
usuario@cliente:~$ ssh usuario@10.0.1.8
Linux wheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.35-2 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  7 20:08:46 2013 from 10.0.1.5
usuario@servidor:~$
```

¿Magia o agujero de seguridad? Pues ni una cosa ni otra, el usuario se ha autenticado de forma segura presentando al servidor sus credenciales a través de una clave RSA de 2048 bits. Podemos comprobar que se ha creado en la cuenta del usario en el equipo servidor el fichero ~/.ssh/authorized_keys y en él se ha agregado la clave pública id_rsa.pub en una línea:

```
usuario@servidor:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCW5qF7fV4AsEh42/LhwzSW879DqJy1+B+YnjGTMlegl2P1QCgdL2E6UNR
TAT3VfsmNpff+gvgvSZSDN8cytUAvV9GrCKn8jnkp9GsxbyN6t4TOlFbUr2dpm6NKwgxdMqWeDWPDNZkwm04KrBEea7K3ns
rGvFuZNzOkQRM9MV+/QldDPY3p/WpUlF8w1j1eWN+T1tFAvKrGy6nQ1giRfUhflUBOH098yrCO0rxDinKKDqkd8T9WgxrpA
yA2Le4LeG9rR8csfZ4p1G6kD5mVY741Tv7X3GifZtQhF8+uMdjxPVC4mH5dJVbORCIJSaljdop00kAci7VsmkqfDH1B5os7
 usuario@cliente
```

Este método es perfecto para sesiones no interactivas (programas automáticos) e incluso es muy cómodo para uso habitual, pero es evidente que tiene un importante riesgo de seguridad que hay que contemplar antes de utilizarlo de forma generalizada: cualquiera que tenga acceso a la clave privada de un equipo podrá acceder sin limitaciones a las cuentas del usuario en todos los equipos en los que se haya exportado la clave pública, si además esto lo hiciera un usuario privilegiado los riesgos en los equipos remotos serían considerables. Se debería restringir este método a

entornos muy controlados o usuarios con pocos privilegios en los que exista poco riesgo de daños. Acceso por ssh con clave pública RSA con frase de paso

Acceso por ssh con clave pública RSA con frase de paso

La forma elemental de evitar los problemas del método anterior es proteger la clave privada con una frase de paso, de manera que se utilice esta clave para autenticarse en el equipo remoto pero no pueda utilizarla nadie que no conozca esta frase de paso. Volvemos a generar el par de claves pública/privada, pero ahora ponemos una frase de paso y modificamos el nombre:

```
usuario@cliente:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/usuario/.ssh/id_rsa):
/home/usuario/.ssh/id_rsa2
Enter passphrase (empty for no passphrase): <--- TECLEAMOS LA FRASE DE PASO
Enter same passphrase again: <--- TECLEAMOS LA FRASE DE PASO
Your identification has been saved in /home/usuario/.ssh/id_rsa2.
Your public key has been saved in /home/usuario/.ssh/id_rsa2.pub.
The key fingerprint is:
19:b4:cd:05:0e:bf:a0:51:83:c5:39:0b:70:61:3d:70 usuario@cliente
The key's randomart image is:
+--[ RSA 2048]----+
|        . . .   .|
|       o . . . ..|
|        o     =.+|
|       .   . o X.|
|        S   o E o|
|         =   . * |
|        o .   o .|
|         .     . |
|                 |
+-----------------+
```

NOTA: Para ver este caso de forma más clara es recomendable borrar la clave pública anterior en el equipo servidor (fichero ~/.ssh/authorized_keys).

Copiamos la nueva clave pública RSA al servidor:

```
usuario@cliente:~$ ssh-copy-id -i ~/.ssh/id_rsa2.pub usuario@10.0.1.8
usuario@10.0.1.8's password:
Now try logging into the machine, with "ssh 'usuario@10.0.1.8'", and check in:

~/.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Comprobamos ahora de nuevo el acceso con la clave pública, pero ahora se nos pide la frase de paso para poder utilizar la clave privada:

```
usuario@cliente:~$ ssh -i ~/.ssh/id_rsa2 usuario@10.0.1.8
Enter passphrase for key '/home/usuario/.ssh/id_rsa2': <--- FRASE DE PASO, NO CONTRASEÑA
Linux wheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.35-2 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  7 21:13:58 2013 from 10.0.1.5
usuario@servidor:~$
```

Es el método más seguro de todos, ya que para poder acceder al sistema remoto necesitamos la clave privada del usuario y conocer la frase de paso, es este método el que se utiliza en determinados sistemas deshabilitando el acceso por ssh con contraseña. El inconveniente de este método es que hay que teclear la frase de paso para cada conexión y en determinadas ocasiones esto puede ser realmente pesado, pero ssh-agent viene al rescate.

Acceso por ssh con clave pública RSA con frase de paso y ssh-agent

ssh-agent en un programa que almacena las claves privadas y las utiliza en cada sesión ssh que establezcamos en la sesión actual, lo ejecutamos y añadimos la clave id_rsa2:

```
usuario@cliente:~$ ssh-agent /bin/bash
usuario@cliente:~$ ssh-add ~/.ssh/id_rsa2
Enter passphrase for /home/usuario/.ssh/id_rsa2: <--- TECLEAMOS LA FRASE DE PASO
Identity added: /home/usuario/.ssh/id_rsa2 (/home/usuario/.ssh/id_rsa2)
```

Ahora podemos establecer la conexión ssh sin tener que teclear la frase de paso en cada ocasión, pero sin el peligro de utilizar una clave sin frase de paso ya que un usuario que tuviera acceso a nuestra clave privada no podría utilizar nuestra sesión de ssh-agent:

```
usuario@cliente:~$ ssh -i ~/.ssh/id_rsa2 usuario@10.0.1.8
Linux wheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.35-2 x86_64

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

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  7 21:19:07 2013 from 10.0.1.5
usuario@servidor:~$
```

Es importante destacar que habitualmente ya existe un ssh-agent ejecutándose en el entorno gráfico de una máquina GNU/Linux por lo que en muchos casos no es necesario ejecutarlo, basta con añadir las claves con ssh-add.