SSH - авторизация по ключам

Опубликовано muff в Сб, 2012-07-07 15:47

Одним из самых надежных способов авторизации является авторизация по ключам. Само собой подразумевается, что приватные ключи хранятся в надежном месте... Еще одним плюсом использования авторизации по ключам является возможность использоватния в скриптах.

Рассмотрим пример настройки авторизации по ключам на **FreeBSD**. Создадим групу для удаленных пользователей и непосредственно пользователя **remoteuser1**, используя свободные **uid** и **gid**:

```
test.muff.kiev.ua# pw groupadd -n remoteusers -g 1500
test.muff.kiev.ua# adduser
Username: remoteuser1
Full name: RemoteUser1
Uid (Leave empty for default): 1500
Login group [remoteuser1]: remoteusers
Login group is remoteusers. Invite remoteuser1 into other groups? []:
Login class [default]: russian
Shell (sh csh tcsh nologin) [sh]: tcsh
Home directory [/home/remoteuser1]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : remoteuser1
Password   : *****
Full Name  : RemoteUser1
Uid        : 1500
Class      : russian
Groups     : remoteusers
Home       : /home/remoteuser1
Shell      : /bin/tcsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (remoteuser1) to the user database.
Add another user? (yes/no): no
Goodbye!
```

Переключаемся на созданного пользователя:

```
test.muff.kiev.ua# su remoteuser1
```

Теперь немного теории. Воспользовавшись командой **ssh-keygen** можно создать ключи **DSA** или **RSA**, которыми пользователи могут аутентифицироваться. Согласно материала из Википедии, имеем такие определения:

- **DSA** (Digital Signature Algorithm) [1] — алгоритм с использованием открытого ключа для создания электронной подписи, но не для шифрования (в отличие от RSA и схемы Эль-Гамаля). Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность. Алгоритм основан на вычислительной сложности взятия логарифмов в конечных полях.

- **RSA** (буквенная аббревиатура от фамилий Rivest, Shamir и Adleman) [2] — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи.

Каким из методов воспользоваться - выбирать Вам. Однако стоит отметить особенность алгоритма **DSA** - длина ключа, который генерируется, составляет **1024 бита**. При попытке увеличить размер ключа, в ответ вываливается сообщение "**DSA keys must be 1024 bits**". Однако, стоит отметить тот факт, что **при увеличении размера ключа, возрастает восприимчивость алгоритмов к определенным видам атак**!

Сгенерируем ключ, принимая значения по умолчанию (поле passphrase оставляем пустым).

Генерирование **DSA** ключа:

```
test.muff.kiev.ua% ssh-keygen -t dsa


Generating public/private dsa key pair.Enter file in which to save the key (/home/re
moteuser1/.ssh/id_dsa):Created directory '/home/remoteuser1/.ssh'.Enter passphrase (
empty for no passphrase):Enter same passphrase again:Your identification has been sa
ved in /home/remoteuser1/.ssh/id_dsa.Your public key has been saved in /home/remoteu
ser1/.ssh/id_dsa.pub.The key fingerprint is:d0:f1:a6:e8:d8:d7:3d:2f:90:e6:f8:78:75:8
d:a4:b9 remoteuser1 [at] test [dot] muff [dot] kiev [dot] ua
The key's randomart image is:+--[ DSA 1024]----+|      .      ||     . o     ||    . . o    ||     o o   .
||     . S  . + o ||    +   .+.+ o .||    . o .+.ooo   ||      ...o Eo   ||       .o.  ..  |+-----------------+
```

Генерирование **RSA** ключа, длиной 2048 бит:

```
test.muff.kiev.ua% ssh-keygen -t rsa -b 2048


Generating public/private rsa key pair.Enter file in which to save the key (/home/re
moteuser1/.ssh/id_rsa):Enter passphrase (empty for no passphrase):Enter same passphr
ase again:Your identification has been saved in /home/remoteuser1/.ssh/id_rsa.Your p
ublic key has been saved in /home/remoteuser1/.ssh/id_rsa.pub.The key fingerprint is
:2b:b2:56:42:1f:a6:dd:b6:98:cf:5c:17:a0:f5:01:70 remoteuser1
 [at] test [dot] muff [dot] kiev [dot] ua
The key's randomart image is:+--[ RSA 2048]----+|      ..E     ||      . .    ||       o .   ||   . o  o o .
||   . = oS   o    ||    o + o.   .   ||    .o.+... .    ||    .oo+.. .    ||   ..  .+      |+-----------------+
```

После использования команды **ssh-keygen**, будет созданы пара из публичного и приватного ключей, используемых для аутентификации. Приватный ключ сохраняется в **~/.ssh/id_dsa** (или **~/.ssh/id_rsa** соответственно), а публичный в **~/.ssh/id_dsa.pub** (или **~/.ssh/id_rsa.pub** соответственно). В целях безопасности советую приватные ключи сразу перемещать в надежное место и удалять их с сервера.

Включаем авторизацию по ключам. В файл **/etc/ssh/sshd_config** внесем такие строки:

```
# Разрешение использования RSA ключей
RSAAuthentication yes
# Разрешение авторизации при помощи ключей
PubkeyAuthentication yes
# Путь к ключам, с которыми можно соединяться.
AuthorizedKeysFile .ssh/authorized_keys
```

После внесения изменений перезапускаем демон **sshd**:

```
test.muff.kiev.ua# sh /etc/rc.d/sshd restart
```

Для авторизации по ключам, на удаленном компьютере публичный ключ должен быть помещен в файл **~/.ssh/authorized_keys** . Соответственно, необходимо записать сгенерированные ключи в **~/.ssh/authorized_keys**.

Для **DSA**:

```
test.muff.kiev.ua% cd ~/.ssh/
test.muff.kiev.ua% cat id_dsa.pub >> authorized_keys
```

Для RSA:

```
test.muff.kiev.ua% cd ~/.ssh/
test.muff.kiev.ua% cat id_rsa.pub >> authorized_keys
```

Теперь скопируем приватный ключ на сервер, с которого будем подключаться, в домашний каталог пользователя и в целях безопасности удалим приватный ключ.

DSA:

```
test.muff.kiev.ua% cd ~/.ssh/
test.muff.kiev.ua% scp id_dsa remoteuser1 [at] server [dot] muff [dot] kiev [dot]
ua:/home/remoteuser1/.ssh/
test.muff.kiev.ua% rm id_dsa
```

RSA:

```
test.muff.kiev.ua% cd ~/.ssh/
test.muff.kiev.ua% scp id_rsa remoteuser1 [at] server [dot] muff [dot] kiev [dot]
ua:/home/remoteuser1/.ssh/
test.muff.kiev.ua% rm id_rsa
```

На этом настройка авторизации по ключам заканчивается. Пытаемся залогиниться:

```
server.muff.kiev.ua% ssh test.muff.kiev.ua
test.muff.kiev.ua%
```

Все работает, авторизацию не запрашивает. Это в случае, если будете коннектиться с **Unix** -системы.

Если же попытаться подключиться с **Windows**-системы, используя как **ssh**-клиента утилиту **PuTTY**, то при попытке использовать полученный приватный ключ получим ошибку "**Unable to use key file (OpenSSH SSH2 private key)**".

Для решения этой проблемы необходимо воспользоваться утилитой **puttygen**. Запускаем утилиту и импортируем наш приватный ключ: **Conversions -> Import Key**. После этого сохраняем полученый **ppk**-файл, нажав на кнопку "**save private key**".

Запускаем **PuTTY** и в настройках сессии указываем следующие настройки и действия:

- Connection -> data -> autologin username =User
- Connection -> data ->ssh -auth = <path_to_ppk-file>
- Сохраняем сессию

Запускаем **PuTTY** выбираем сохраненную сессию, нажимаем кнопку "**Open**" и попадаем в систему.

Источник (получено *2025-05-09 22:08*):
http://muff.kiev.ua/content/ssh-avtorizatsiya-po-klyucham

Ссылки:
[1] http://ru.wikipedia.org/wiki/DSA
[2] http://ru.wikipedia.org/wiki/RSA