How to Install Your Public Key

Using the TeraGrid User Portal from a Linux Host

P. A. Cheeseman, (aai@purdue.edu)

Note that these directions are written under the assumption that your *home system*, the system from which you plan to log in to other hosts, allows you to log in via *password* and/or *keyboard interactive* authentication.

First, a bit about SSH2 public keys.

Public keys are generated in two forms, one is referred to as the *OpenSSH key file*, created by *ssh-keygen*, which has a *single line* containing the key. The OpenSSH file contents will look something like …

```
ssh-rsa  YYYYQ7NxN…uK+0Ko58=  yourlogin@yourhost
```

where the third item (*yourlogin@yourhost*) may or may not appear. The above example is abbreviated and the embedded periods refer to many characters which were omitted.

The second form is the *SECSH Public Key File Format* which is created by other SSH tools including *PuTTYgen* and *SecureCRT*. The SECSH form is several lines and resembles …

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20080702"
YYYYQ3NzaC1yc2EYYYYQJQYYYIEY5CRGNRZ2XVOcGiaEgRiIaZPcffmYdKPgpK44
zZ/q7plHY/Fqfzr7Dh5tPLuOF3S7vYq57a2o8TJw3mnF6CmsvlYlYYSYs7Kp3YWm
SE6uQk76yKVQ0C7hCiwheQmGunRY0KlsfGZTfs1rdxnVTQgLiZO2P7leyvru9Upu
K+0Ko58=
---- END SSH2 PUBLIC KEY ----
```

which is more human readable.

The Linux systems at Purdue recognize the OpenSSH form of public keys. You must know the form of your public key before you install it.

The steps below are meant to describe what you do after you have generated a public/private key pair. Typically, the public key will be saved in a file with a suffix of *pub*, for instance 'id_rsa.pub' or 'identity.pub' in a location specific to your SSH2 client. For Linux systems, key pairs are stored, by default, in directory $HOME/.ssh, where $HOME refers to your home directory.

How you generate your public/private key pair, by the way, is an issue that involves the SSH2 client you're using. Little is said here about key pair generation because methods of key pair generation are already documented for the various clients. If you are new to SSH, however, make sure you do not confuse your public key with your *private key*.

Now, a recipe for installing your public key *for the first time* ...

1. Log in to the *remote host*, i.e. the host where you plan to install your public key, via the TeraGrid User Portal.

2. Once logged in, create a directory named '.ssh' in your home directory by entering the command

   ```
   mkdir  $HOME/.ssh
   ```

3. You should then be able to do a remote copy of the public key from your home system to the remote system via …

   ```
   scp  yourlogin@yourhome:publickeypath  $HOME/.ssh/remotename
   ```

   where *yourlogin@yourhome* refers to the login you use in your home system and *publickeypath* is the path to your public key file (e.g. .ssh/id_rsa.pub). For *remotename*, use 'foo' if the public key on your home system is in SECSH format. Otherwise, use 'authorized_keys'.

   In this step, you should be prompted for your password on your home system in order to complete the copy. *Note that any previous authorized_keys file is replaced by this operation or Step 4 below*.

4. If the public key on your home system was in OpenSSH format, skip this step. Otherwise, do the following on the remote system …

   ```
   cd  $HOME/.ssh
   ssh-keygen  –i  –f  foo  >  authorized_keys
   rm  foo
   ```

5. For fastidiousness, you may want to make the entire .ssh directory readable only by you. To do so, enter

   ```
   chmod  -R  go–rwx  $HOME/.ssh
   ```

   which makes the .ssh directory and all of its contents inaccessible to other users on the system.

6. Log out of the remote system and exit the user portal.

7. Try to log in directly to the remote system from your home system using your usual SSH2 client. If this step doesn't work as expected, make sure you've specified public key authentication as your primary authentication method. If things still aren't working, that's why I've included my e-mail address above ☺.

Some final remarks seem important w.r.t. the authorized_keys file. Here they are.

You may achieve login to the remote system from multiple home systems by simply adding the public key for each of your home systems to the authorized_keys file on the remote system. The obvious risk associated with this practice is that you open the remote system to more points of access. The point here is that you need to be aware of what systems are permitted access via the authorized_keys file and keep your authorized_keys tidy. Generally speaking, the authorized_keys file contains a single line entry for each key used to log in from your home systems. Blank lines are ignored so you can space things out in the file to aid with readability.

For more information about the commands used in this note, see *ssh-keygen(1)*, *scp(1)*, and many other manual pages describing the various SSH utilities.

P. A. Cheeseman
aai@purdue.edu