SSH Exercise

We are going to generate a key pair using the ssh client we are using in the workshop. If you are using Linux or Mac OS X this will probably be OpenSSH -- if you are using Windows, this will probably be PuTTY.

1. Create a new public/private key pair

If you are using PuTTY or some other Windows ssh client, hopefully it is intuitively obvious how to generate a key pair. If possible, you shoud set a passphrase, which will protect your key pair on disk.

If you are using OpenSSH:

Making sure that you are not root, use the ssh-keygen command to generate a public/private key pair.

Notice that you can usually accept the suggested filename to store the keys.

You should set a passphrase. We suggest to make things easy while we are practicing that you should use the passphrase "success!".

```
[pc99:~]% ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/afnog/.ssh/id_rsa):
Enter passphrase (empty for no passphrase): <type success!>
Enter same passphrase again:  <type success!>
Your identification has been saved in /home/afnog/.ssh/id_rsa.
Your public key has been saved in /home/afnog/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:cw1ehcODDgsVrxQQ2Yka8ABjRFqGNB2rbMBpCwzkEQs afnog@pc99
The key's randomart image is:
+---[RSA 2048]----+
|E%Bo. oB+o o ..  |
|X+=+o + +o. =.   |
|+B ..o ..+o .o   |
|= o .  ..o.+     |
| =      S o .    |
|.        o       |
|                 |
|                 |
|                 |
+----[SHA256]-----+
[pc99:~]%
```

As you can see from the output produced by ssh-keygen (which of course
you read) the public key is in the file "/home/afnog/.ssh/id_rsa.pub".

You can display it on the screen using the more or cat commands, e.g.

```
$ cat /home/afnog/.ssh/id_rsa.pub
```

2. Copy your public key to your server

Connect to your server using ssh. You will need to use the same password as normal, since we have not yet transferred the ssh public key to the server. Once you have completed this part, you will not need to use the password any more.

The easiest way to copy the public key to the server is to use copy and paste. Copy the public key that you generated on your laptop and paste it into the right file on the server.

2.1. Copy the public key

This part needs to be done on your laptop.

Refer to section 1 for displaying the public key you generated on your screen. Once it is on your screen, select it and copy it to the clipboard.

2.2. Paste the public key into the right place

This part needs to be done on the server command line.

YOU SHOULD NOT BE ROOT FOR THIS PART. Make sure you are user "afnog".

```
$ whoami
afnog
$
```

If you don't already have a directory in your home directory called ".ssh", create one.

```
$ mkdir ~/.ssh
```

If you get an error because the directory already exists, that's ok.

Edit the file "authorized_keys" in that directory.

```
$ cd ~/.ssh
$ vi authorized_keys
```

You can use nano if you like, instead of vi.

This file contains a list of public keys that are used to identify clients
who can get access to this account. The list can have more than one public

key on it (you might already have one entry in there).

Add your public key to the end of the file by pasting it from the clipboard.

Save the file.

3. Check that you can connect without a password

Disconnect from your server.

Connect again. Your ssh client should ask you for a passphrase for your
public key (we specified "success!" when we generated the key pair).

The server should not ask you for a password.

4. Disable password authentication on the server

Since we are now able to authenticate using our private keys, we no longer need to use passwords. It is good security practice to disable things we don't need, so let's disable password authentication.

(This is a effective way to stop worrying about ssh scanners on the Internet. They can try whatever passwords they want, but they will never get in.)

As root, edit the file /etc/ssh/sshd_config

```
  # vi /etc/ssh/sshd_config
```

Find this part of the file:

```
  # Change to no to disable tunnelled clear text passwords
  #PasswordAuthentication yes
```

and un-comment the second line, and change the option to "no". It should look like this:

```
  # Change to no to disable tunnelled clear text passwords
  PasswordAuthentication no
```

Now restart the ssh server:

```
  # service sshd restart
  #
```

Disconnect and confirm that you can still connect using ssh (repeat section 3 above).