Setup ssh-key exchange

It is possible to automatically login in a ssh session on a remote system, without entering a password. Also copying files with scp without password is possible. To make this possible a public ssh key needs to be exchanged between the source and the target system. On the source system an ssh key pair is generated. The public key is transferred to the target system and will be used to validate the login attempt that is signed with the private key.

Key Generation

On the Source System

The next commands should be performed by the user that wants to ssh/scp to the target system. Steps below assume that this is the root user, but this could also be www-data, etc. As user on the source system, use ssh-keygen to generate a public/private key pair.As a password, you would type nothing (just enter). This will save the public key in /root/.ssh/id_rsa.pub and the private key in /root/.ssh/id_rsa, if you don't specify another location.

```
# cd /root/.ssh
# ssh-keygen -t rsa -b 2048
-------------------------------------------------
# Generating public/private rsa key pair.
  Enter file in which to save the key (/root/.ssh/id_rsa):
  Enter passphrase (empty for no passphrase):
  Enter same passphrase again:
  Your identification has been saved in /root/.ssh/id_rsa.
  Your public key has been saved in /root/.ssh/id_rsa.pub.
```

Key Exchange

To allow the client to ssh to the target system as root, you need to place the client's public key into root's authorized list on the target system. There are 2 different ways to achieve this:

- Manual
- With 'ssh-copy-id'

Key transfer - Manual

Append client's public key (BackupPC_id_rsa.pub) to root's /root/.ssh/authorized_keys2 file on the client:

On the Target System

Get the public key from the client system and add it to the 'authorized keys':

```
# cd /root/.ssh
# scp oscar@192.168.xx.xx:/home/oscar/.ssh/id_rsa.pub  client_id_rsa.pub
# touch ~/.ssh/authorized_keys2
# cat client_id_rsa.pub >> ~/.ssh/authorized_keys2
# rm client_id_rsa.pub
```

On the Client System

You need to place the target's public key into client's ~/.ssh/known_hosts file, otherwise you will get a “Host key verification failed.” error, and the client will not be able to log into the target system. To place the target public key into the ~/.ssh/known_hosts file, you need to do this (make sure ~/.ssh/target_id_rsa.pub is the client's public key, which needs to be copied from the target system):

```
$ scp root@192.168.xx.xx:/root/.ssh/id_rsa.pub target-key.pub
$ touch ~/.ssh/known_hosts
$ cat ~/.ssh/target-key.pub >> ~/.ssh/known_hosts
$ rm target-key.pub
```

Or symply try to ssh from client to target as root. The key will be placed automatically in known_hosts file.

Key transfer - with ssh-copy-id

Copy your keys to the target system:

```
$ ssh-copy-id -i id_rsa.pub root@targetsystem

remoteusername@targetsystem's password:
```

Now try logging into the machine, with ssh 'remoteusername@targetsystem'. The key of your system should now be place in to the .ssh subdirectory in the home directory of remoteusername on the target system.

```
/home/remoteusername/.ssh/authorized_keys
```

or

```
/root/.ssh/authorized_keys
```

or on openwrt:

```
/etc/dropbear/authorized_keys
```

Test

This should now work from the server to the client:

```
$ ssh root@192.168.xx.xx
```

If everything went ok, you should be logged in directly without a password prompt.