

SSH to another machine (Linux, openSUSE)

SOP

Ruediger Teichert
ruediger@teichert-online.de

v0.1

November 2009

Introduction

- In a network, we want to use the terminal (shell) to get root rights on another machine. In other words, we can use the shell of computer 1 to work as root on computer 2.

Prerequisites

- Two computers with network cards and Linux installed, for this manual one machine has openSUSE 11.1 and one has openSUSE 11.2 (32 bit). Both were connected to ADSL network.
- GUI (KDE) installation is assumed.
- For first try it is assumed both computers have their firewalls disabled.
- Assumed: one machine is called linux-box1 and has openSUSE 11.2 installed and the other is linux-box2 with openSUSE 11.1 installed. Version 11.2 has ssh disabled by standard.

Step 1, ssh installed?

- *We need SSH on both machines. In case SSH had not been installed, install it now on the server and all clients. You can use Yast to install ssh.*

Step 2: Configure ssh

- On the first machine we need to generate an ssh key (for secure shell communication between the computers):

```
[root@linux-box1]:# ssh-keygen -t dsa
```

Note: always only type what is behind the #

This will produce lots of output and asks some questions. Just press ENTER when being asked:

```
Generating public/private dsa key pair.
```

```
Enter file in which to save the key (/root/.ssh/id_dsa):
```

```
Created directory '/root/.ssh'.Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /root/.ssh/id_dsa. Your public key has been saved in /root/.ssh/id_dsa.pub. The key fingerprint is:d7:ad:ce:0f:e2:d8:16:3d:92:10:b8:b3:7f:7b:68:d9
```

Step 2 (cont.)

- `[root@linux-box1]:# cd /root/.ssh/`
- `[root@linux-box1]:# cp ./id_dsa.pub ./authorized_keys`
- Now we copy this `/root/.ssh/` directory to file system of the peer `linux-box2`.
- To do this, we first need to make sure, the packages `limal-nfs-server` and `nfs-client` and `yast2-nfs-client`, `yast2-nfs-common` and `yast2-nfs-server` are installed on both machines.

Step 2 (cont. II)

- We login to linux-box2 (the other computer!) now and create a directory /export being root:
[user1@linux-box2]:# `su root`
[root@linux-box2]:# `mkdir /export`
- We enable NFS-Server via yast and export the /export directory.
- Then again on linux-box1 we create an import directory called /import.
[user1@linux-box1]:# `su root`
[root@linux-box1]:# `mkdir /import`
- We go into the NFS client configuration under Yast's network services and add linux-box2 as NFS server (you can use its IP address to be sure). We then select the remote directory "export" and select "/import" as local mount point.

Step 2 (cont. III)

- Now we can copy the .ssh directory into the file system of the 2nd computer:

```
[root@linux-box1]:# cp -r /root/.ssh /import
```

- This is writing the files to /export on linux-box2 (it is mounted as /import on linux-box1). On box2 we copy the files to root:

- [root@linux-box2]:# cp -r /export/.ssh /root

Step 3: connect

- Establish ssh connections to linux-box2 from linux-box1:
While connecting to the other computer for the first time, you need to enter 'yes'.
- [root@linux-box1]:# `ssh root@linux-box2`

You will be notified “have a lot of fun” and can work as root on linux-box2 now. You will notice linux-box2 in front of your prompt now!

You do this on linux-box1: [root@linux-box2]:# `exit`

The “exit” will leave the shell of the remote machine.

Then repeat this from linux-box2...

Step 3 (cont.)

```
[root@linux-box2]:# ssh root@linux-box1
```

However you will not succeed if openSUSE 11.2 or a later version is installed on linux-box1. At least the verbose version “`ssh -v root@linux-box1`” will reveal port 22 getting refused by linux-box1.

Reason is, starting with openSUSE 11.2, ssh is by standard disabled.

To solve this, use Yast and Click on System on the left hand side and “System Services (Runlevel)”. Start the “sshd” service in the services overview.

Finally leave with “`exit`”.

Step 4: Firewalls

- If Firewalls are enabled, you need to allow the Services (under “Allowed Services”) ‘NFS Client’ and ‘NFS Server Services’ plus ‘NIS Client’ and ‘Secure Shell Server’ to allow for both the NFS data exchange and the ssh communication.
- Also, under “Custom Rules”, add an entry where you specify the Source network (I was in the 192.168.102.X network, technically called 192.168.102.0 network, so I specified 192.168.102.0), choose protocol TCP and destination port 22 (no source port).