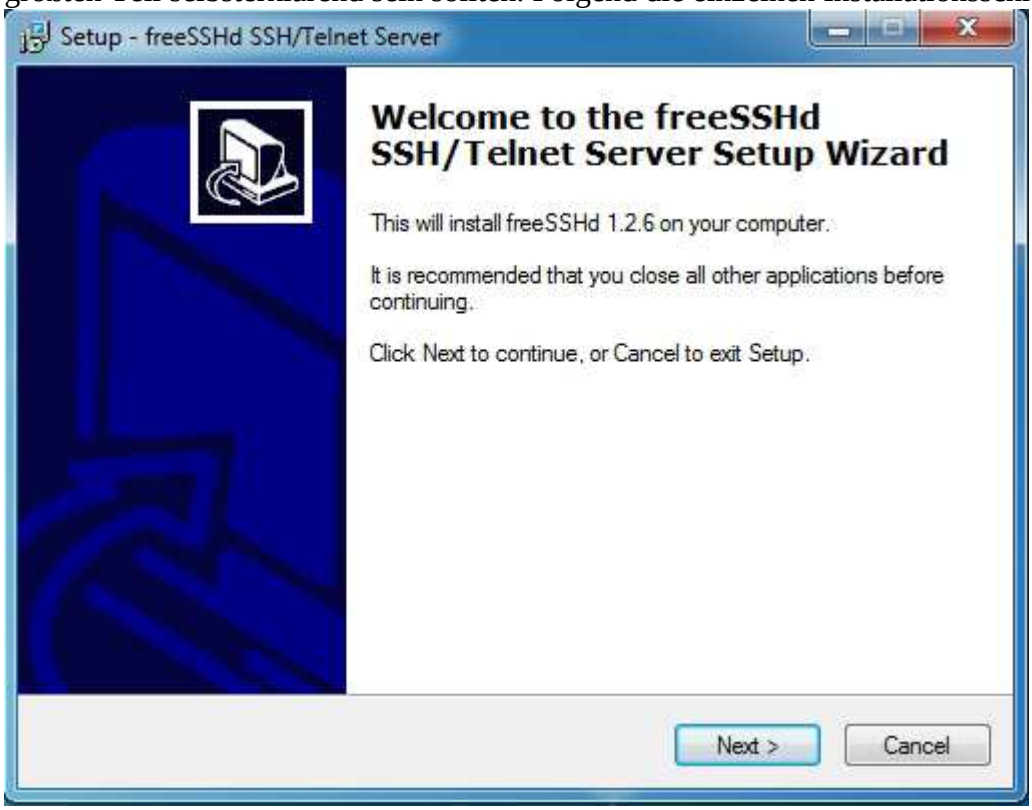
How to install freeSSHd

Enthaltene Funktionen

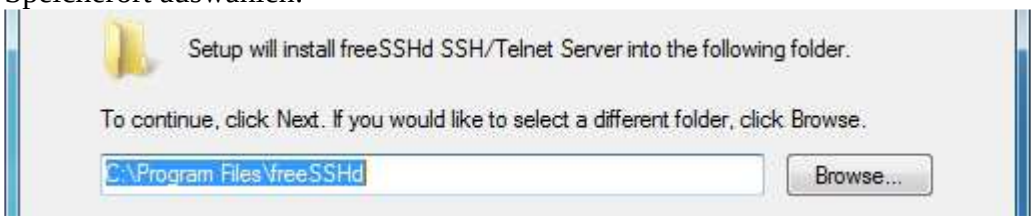
- **Installation**
- **Benutzer anlegen**
- **Verbindung testen**

1. Installation von freeSSHd

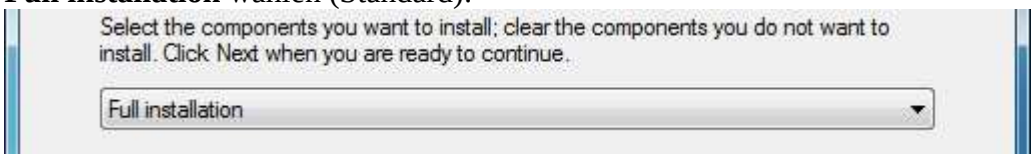
- Falls noch nicht vorhanden, können Sie das Freeware Programm unter folgendem Link downloaden: http://www.freesshd.com/freeSSHd.exe
oder: http://johann-scharl.de/tools/freeSSHd.exe
Starten Sie dann das Setup und folgen Sie den Installationsanweisungen, wobei diese zum größten Teil selbsterklärend sein sollten. Folgend die einzelnen Installationsschritte:



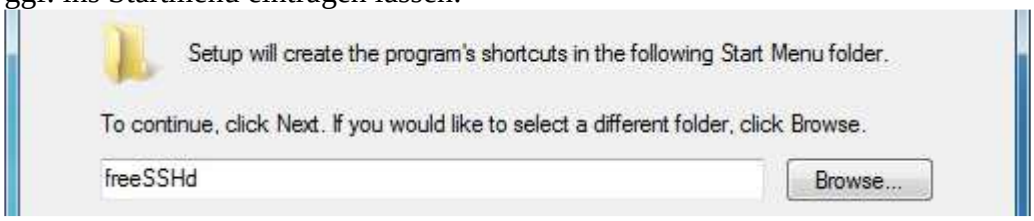
Speicherort auswählen:



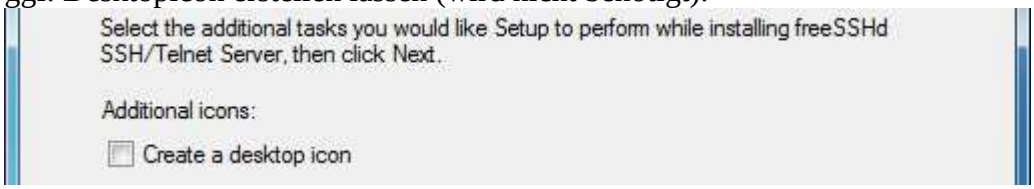
Full installation wählen (Standard):



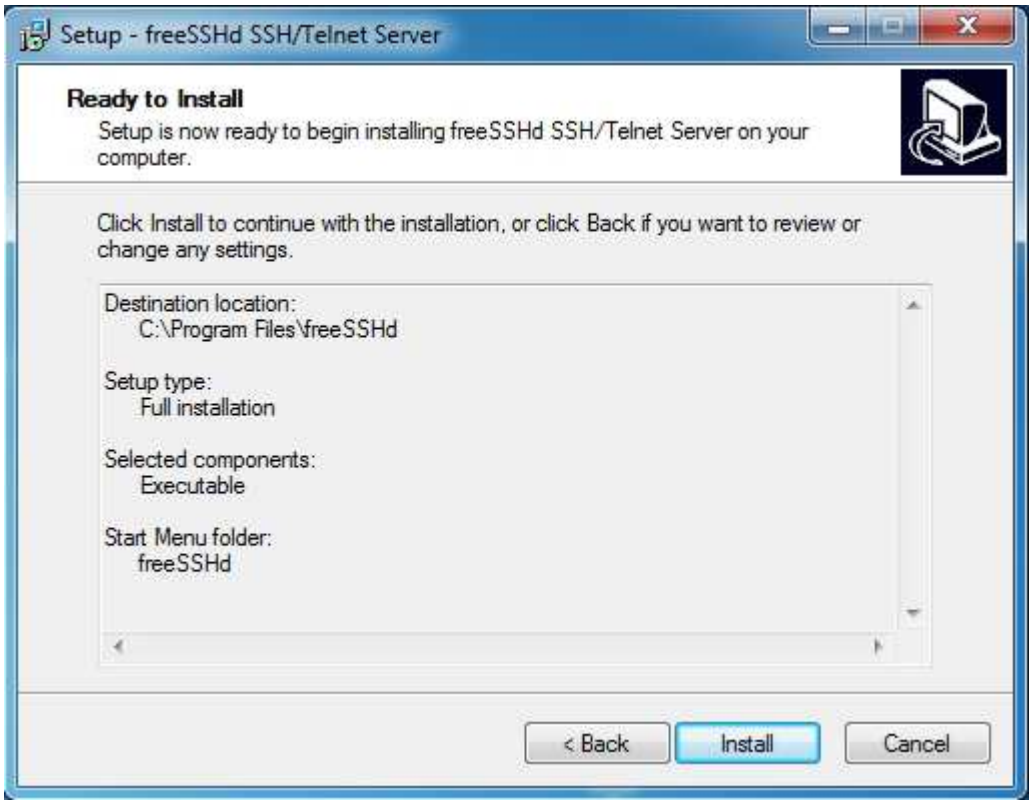
ggf. ins Startmenü eintragen lassen:



ggf. Desktopicon erstellen lassen (wird nicht benötigt):

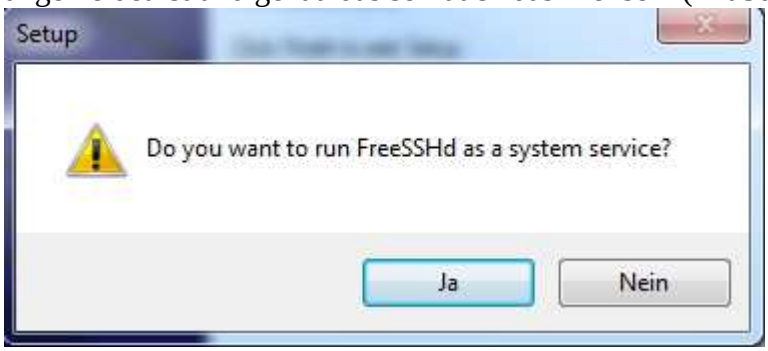


Installation starten:

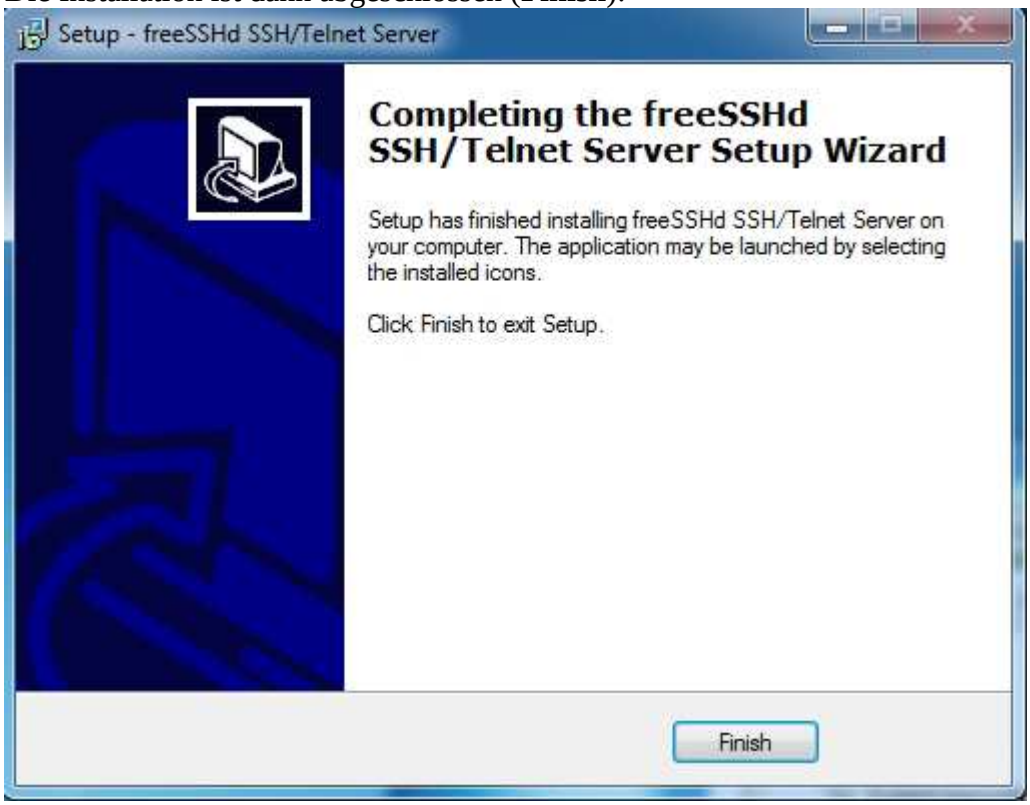


freeSSHd erstellt auf Wunsch in diesem Schritt Private Schlüssel für verschlüsselte Verbindungen. Sie können diese erstellen lassen (mit **Ja** bestätigen), jedoch wird in diesem Tutorial eine unverschlüsselte Verbindung gezeigt. Wie eine verschlüsselte Verbindung erstellt wird, erfahren Sie dann im Tutorial:
http://johann-scharl.de/instructions/How to Public key authentication with freeSSHd.pdf

freeSSHd kann als Systemdienst installiert werden. Dies sollte auch immer so gemacht werden, da der Dienst dann auch erreichbar ist, wenn kein User an Windows angemeldet ist und genau das soll auch das Ziel sein (mit **Ja** bestätigen):



Die Installation ist dann abgeschlossen (**Finish**):



Nachdem die Installation beendet ist, prüfen wir, ob ein neuer Dienst eingetragen wurde. Öffnen Sie die Dienste aus der Systemsteuerung und suchen nach: FreeSSHDService

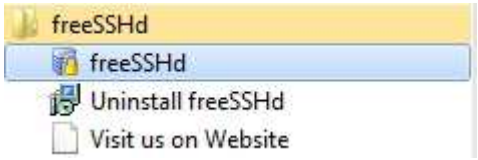
Der Dienst sollte gestartet sein und somit „hört“ das System nun auf den Port 22.
Info: Sollte der Dienst nicht gestartet sein, so versuchen Sie den Start manuell. Verweigert der Dienst den Start, so ist bereits eine andere Anwendung gestartet, welche den gleichen Port benutzt. Um dieses Problem zu lösen, müssen Sie entweder freeSSHd oder der zweiten Anwendung einen anderen Port zuweisen.

Anhand netstat können Sie feststellen, ob der Port 22 abgehört wird. Hierfür starten Sie die Eingabeaufforderung und tippen folgenden Befehl ein: netstat -an

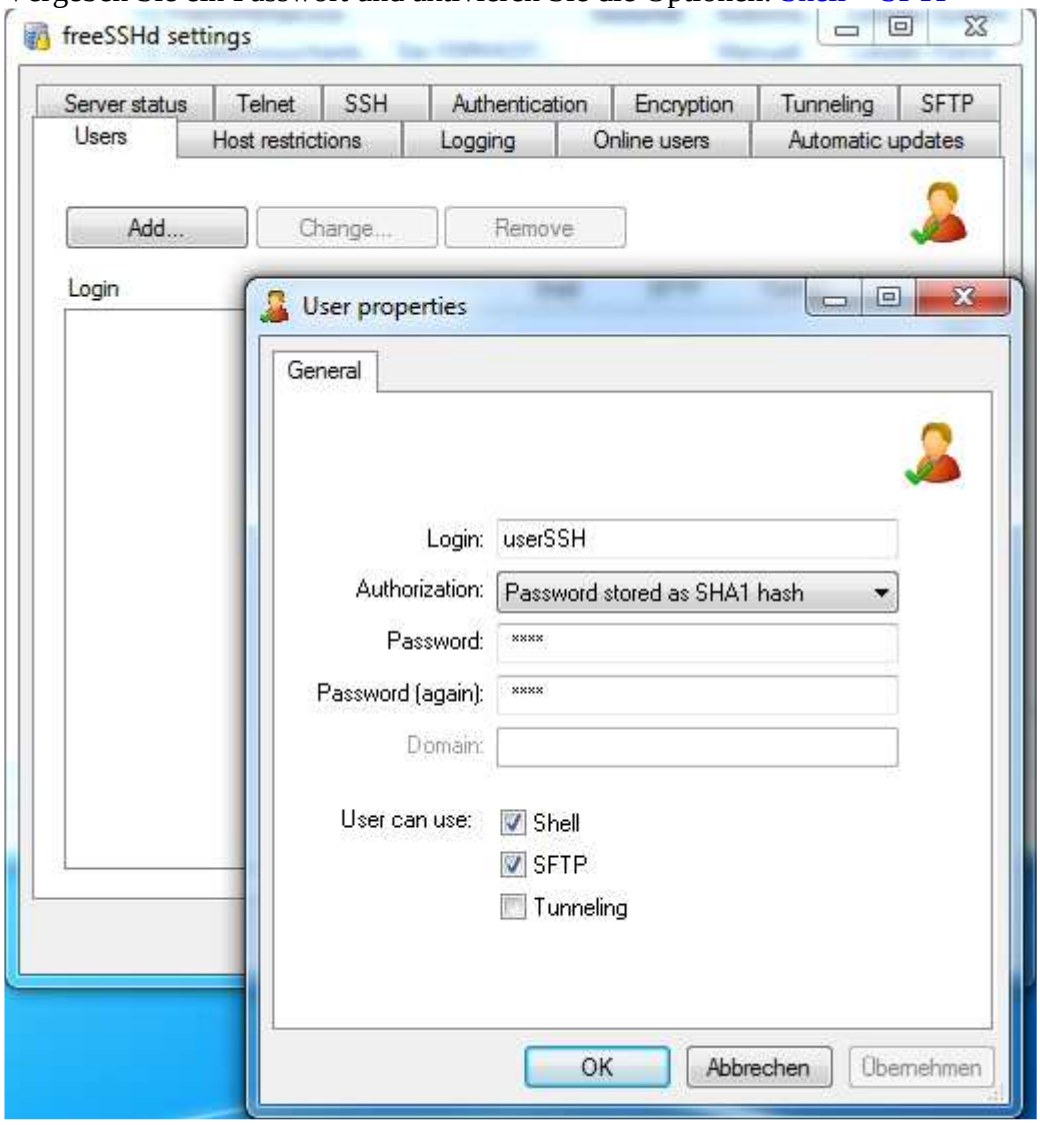
Es erscheint eine Liste mit mehreren Socket-Adressen unter welchen auch die Adresse mit dem Port 22 aufgelistet sein sollte (im Beispiel: 0.0.0.0:22 „ABHÖREN“):

```
C:\Users\win7>netstat -an

Aktive Verbindungen

  Proto  Lokale Adresse          Remoteadresse          Status
  TCP    0.0.0.0:22             0.0.0.0:0              ABHÖREN
  TCP    0.0.0.0:80             0.0.0.0:0              ABHÖREN
  TCP    0.0.0.0:135            0.0.0.0:0              ABHÖREN
```

Das System „hört“ also nun auf den Port 22 und eine Verbindung sollte auch schon möglich sein, nachdem wir im nächsten Schritt einen Benutzer angelegt haben, welcher auch die Verbindung nutzen darf. Hierfür rufen wir das Startmenü von Windows auf und starten die Oberfläche von freeSSHd:

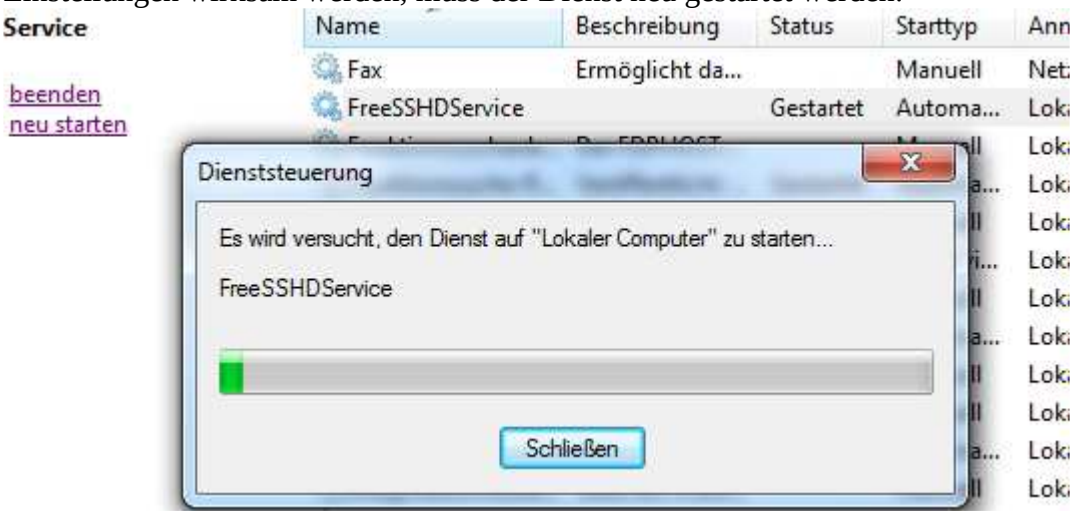


Das Programm wird geöffnet und wir wechseln dann in den Reiter <Users>.
Jetzt fügen wir mit Add… einen neuen Benutzer hinzu. Im Beispiel wird ein Benutzer mit den Namen userSSH angelegt. **WICHTIG**! Authorization: Passwort stored as SHA1 hash
Vergeben Sie ein Passwort und aktivieren Sie die Optionen: Shell + SFTP



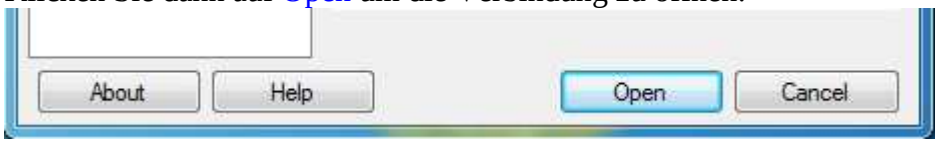
Bestätigen Sie die Eingabe mit OK und der Benutzer erscheint dann in der Liste:

Beenden Sie dann die Einstellungen mit OK. **WICHTIG!** Damit nun diese neuen Einstellungen wirksam werden, muss der Dienst neu gestartet werden:

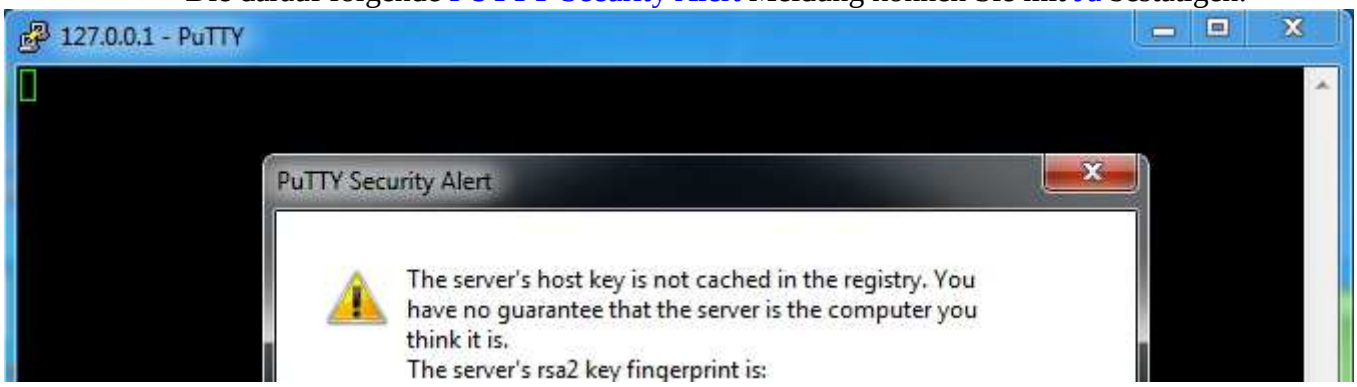


Jetzt können wir die Verbindung mit PuTTY testen. Starten Sie hierfür PuTTY auf dem gleichen Rechner und tippen Sie als Host Name folgende IP-Adresse ein: 127.0.0.1 als Port 22 und Connection type SSH:

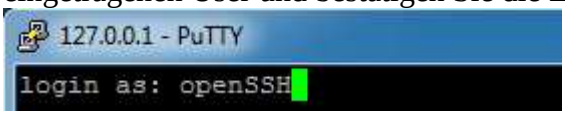
Klicken Sie dann auf Open um die Verbindung zu öffnen:



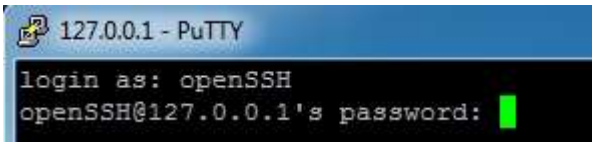
Die darauf folgende PUTTY Security Alert Meldung können Sie mit Ja bestätigen:



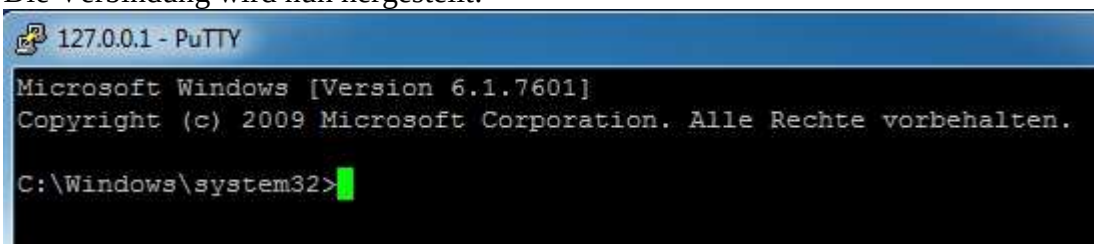
Nun müssen Sie Sich mit einem Usernamen anmelden. Hierfür verwenden Sie den zuvor eingetragenen User und bestätigen Sie die Eingabe mit {ENTER}:



Jetzt geben Sie das dazugehörige Passwort an. Beachten Sie, dass der Cursor bei der Eingabe nichts anzeigt. Lassen Sie Sich hier nicht täuschen, dies dient alleine der Sicherheit. Geben Sie also das Passwort ein und bestätigen Sie wieder mit {ENTER}:



Die Verbindung wird nun hergestellt:



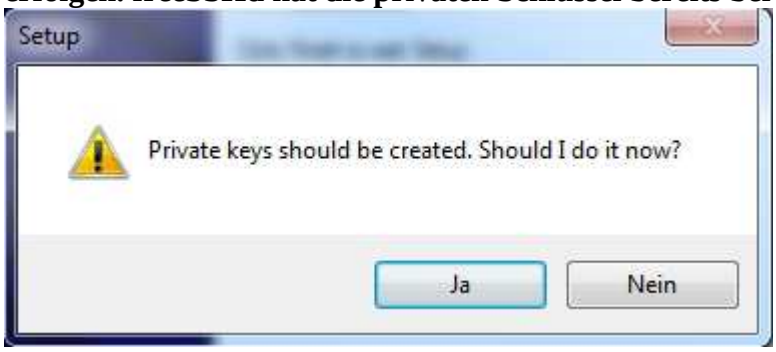
Tippen Sie nun exit ein, um den Modus und die Verbindung zu verlassen. Beachten Sie, dass dies nur ein Test innerhalb des Systems war. Testen Sie jetzt die Verbindung von einem anderen Rechner aus. Starten Sie also PuTTY auf einem anderen Rechner und tragen Sie dann als Host Name die IP-Adresse oder den Rechnernamen des Systems ein, auf welchem freeSSHd installiert wurde. Im Beispiel läuft also der Dienst auf dem System mit der IP-Adresse: 192.168.178.100

WICHTIG! Die Verbindung zum SSH-Server kann nur dann funktionieren, wenn auch die Firewall entspr. konfiguriert ist. In diesem Beispiel muss also der Port 22 als eingehende TCP-Verbindung freigegeben sein.

Es kommt immer wieder vor, dass die Verbindung aufgrund der Firewall-Blockade nicht funktioniert. Putty verhält sich in den meisten Fällen so, dass nach einem Versuch die Verbindung zum Host aufzubauen, folgende Fehlermeldung erscheint: „Network error: Connection timed out“

Die gleiche Fehlermeldung wird auch dann ausgegeben, wenn der Host nicht erreichbar ist, also wenn Sie z.B. eine falsche IP-Adresse oder Rechnernamen angeben oder einfach der Host von außen nicht erreichbar ist.

Info:
Dieses Tutorial beschreibt eine einfache Verbindung via SSH und unverschlüsselt! Eine verschlüsselte Verbindung kann nur mit öffentlichen und privaten Schlüsseln erfolgen. freeSSHd hat die privaten Schlüssel bereits bei der Installation erzeugt:



Im Tutorial: http://johann-scharl.de/instructions/How to Public key authentication with freeSSHd.pdf erfahren Sie dann, wie eine verschlüsselte Verbindung mit PuTTY über SSH hergestellt werden kann.