

Tutorial: Raspberry Pi fernsteuern - SSH und Remote-Desktop

Tutorial: Raspberry Pi mit Hilfe von SSH und XRDP (Remote-Desktop) aus der Ferne bedienen und steuern.

Raspberry Pi - SSH und XRDP Installation, Konfiguration und Troubleshooting.

Wer auch immer, egal wofür, einen oder mehrere Raspberry Pi in seinem Netzwerk betreibt, wird in kürzester Zeit feststellen, wie einfach und bequem sich Einstellungen und Bedienungen an den Rechnern vornehmen lassen, in dem man sie per SSH oder Remote-Desktop-Verbindung fernsteuert.

Aber auch hier hat der liebe Gott den Fleiss vor den Erfolg gestellt, so dass wir an unserem neu eingerichteten Pi noch ein wenig schrauben und justieren müssen, bis das dann funktioniert.

Voraussetzungen schaffen

Grundsätzliche Voraussetzung für eine Fernsteuerung via Netzwerk ist natürlich eine funktionierende Netzwerk(LAN)-Verbindung zwischen dem fernsteuernden und dem fernzusteuerten Rechner.

Für die zu installierenden Pakete benötigen wir auch natürlich wieder einen Internetzugang für unseren Raspberry Pi.

Haltet Euch bei Euren Versuchsaufbauten bitte möglichst immer an die KISS (Keep It Stupid Simple)-Regel. Baut also immer zuerst möglichst einfache Konstrukte, bringt sie zum Laufen und macht sie dann erst komplexer. Für unsere SSH- und XRDP-Versuche heisst dass: Stöpselt am besten den Raspberry Pi in das gleiche Netzwerksegment (gleicher Router) wie Euren Rechner, dann fällt schon einmal das Netzwerk als mögliche Fehlerquelle aus unserem Fernsteuerungsversuch weitestgehend heraus.

Jetzt sollten wir noch die IP-Adresse, die der Raspberry Pi von unserem Router oder von uns selbst zugewiesen bekommen hat, herausfinden und notieren.

Nun fehlt uns nur noch eine Software, die wir für die Fernsteuerung unseres Himbeerküchleins verwenden können. Für Windows-Rechner bietet sich hier das kostenlos im Internet zu findende Tool namens "Putty" an.

Sucht danach einfach auf z.B. der Welt grösster Suchmaschine und ladet es Euch auf Euren Rechner herunter. Achtet dabei vielleicht darauf, dass Ihr es nicht von einer Wald-und-Wiesen-Seiter herunterladet, sondern wählt als Quelle für Euren Download einen soliden Anbieter, wie vielleicht eine bekannte Computerzeitschrift, aus.

Putty muss nicht installiert werden. Kopiert es einfach dorthin, wo Ihr es haben wollt und schon könnt Ihr es durch einen simplen Doppelklick starten.

Damit SSH mit dem Raspberry Pi funktioniert, müsst Ihr es noch in Raspi-Config freischalten. Das geht so:

\$ pi@rechnername: sudo raspi-config (enter)

Hier wählt Ihr aus dem Menü den Punkt 'Interfacing Options' aus.

Im darauffolgenden Menü wählt ihr den Punkt SSH aus und beantwortet die Frage, ob der SSH-Server eingeschaltet (enabled) werden soll mit Ja.

Nach der Beendigung von raspi-config sorgt ein Neustart des Raspberry Pi dafür, dass die geänderten Einstellungen auch übernommen werden. Ein Neustart geht so:

```
$ pi@rechnername: sudo reboot (enter)
```

## Raspberry Pi - SSH installieren

SSH (Secure SHell) muss auf Eurem Linux nicht installiert werden. Da es sich hierbei um eines der ureigensten Werkzeuge zur Steuerung eines Linux-Rechners handelt, ist es als Softwarepaket generell schon vorinstalliert.

Es genügt, den Dienst in der Konfigurationsoberfläche "raspi-config" (Siehe Tutorial Grundeinrichtung) zu aktivieren, noch einmal zu booten und schon kann es losgehen.

Leider habe ich schon oft die Erfahrung gemacht, dass eine spontane SSH-Verbindung zwischen einem Windows-Rechner und einem Raspberry Pi, mangels vorhandener bzw. nicht ordentlich benannter Verschlüsselungs-Chiffren vom Pi abgelehnt wird. Das lässt sich jedoch einfach ändern.

Geht mit Maus, Tastatur und Bildschirm an Euren Pi und loggt Euch ein. Wenn Ihr im Desktop seid, öffnet einfach ein Terminalfenster und schon kann es losgehen.

Wenn wir die IP-Adresse des Raspberry-Pi nicht wissen, beginnen wir damit, sie herauszufinden. Dazu tippen wir in die Konsole folgendes Kommando ein:

```
pi@rechnername: ~ $ ip addr show (enter)
```

An dieser Stelle ist Linux dann so freundlich und zeigt Euch die Netzwerk-Konfiguration Eurer Netzwerkkarten an. Ist euer Raspi per LAN-Anschluss mit dem Netzwerk verbunden, interessiert Euch hier der Eintrag 2: eth0.

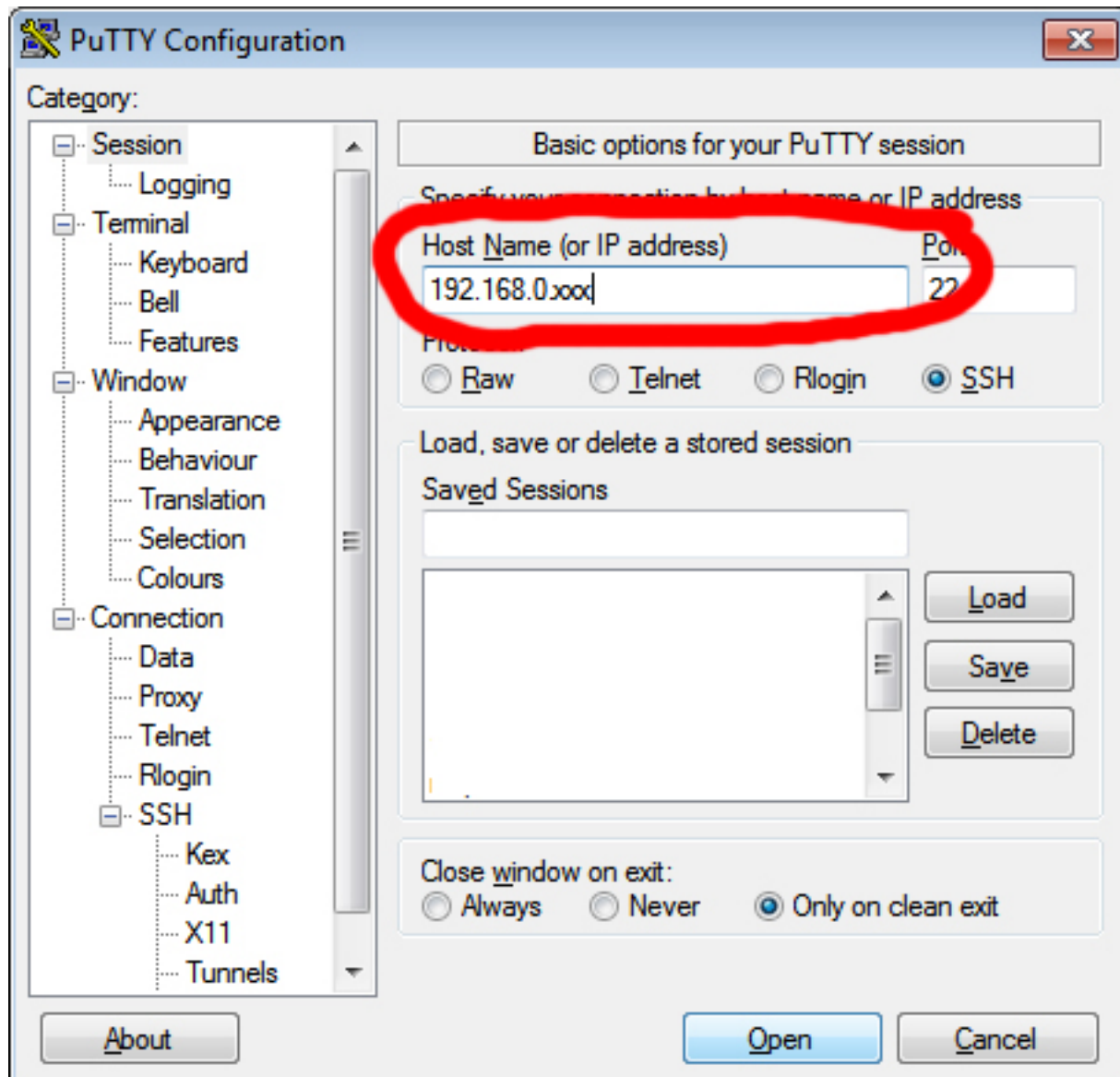
Solltet Ihr die WLAN-Verbindung nutzen, so ist es der Eintrag 3: wlan0.

Die Zeile mit inet (nicht inet6) enthält in der Form xxx.xxx.xxx.xx/xx die IP-Adresse, die Eurer Raspberry Pi im Netz hat. Die IP-Adresse ist quasi der mathematische Name eines Netzwerkgerätes im lokalen Netzwerk.

Auch wenn Euer Pi einen lesbaren Namen (Hostnamen) von Euch bekommen hat, so wird anhand dieses Namens doch von allen anderen Geräten, bevor sie im Netzwerk Euren Pi ansprechen wollen, diese IP-Adresse herausgefunden (das macht z. B. der DNS) und dann wird das Gerät/der Server per IP-Adresse angesprochen und adressiert.

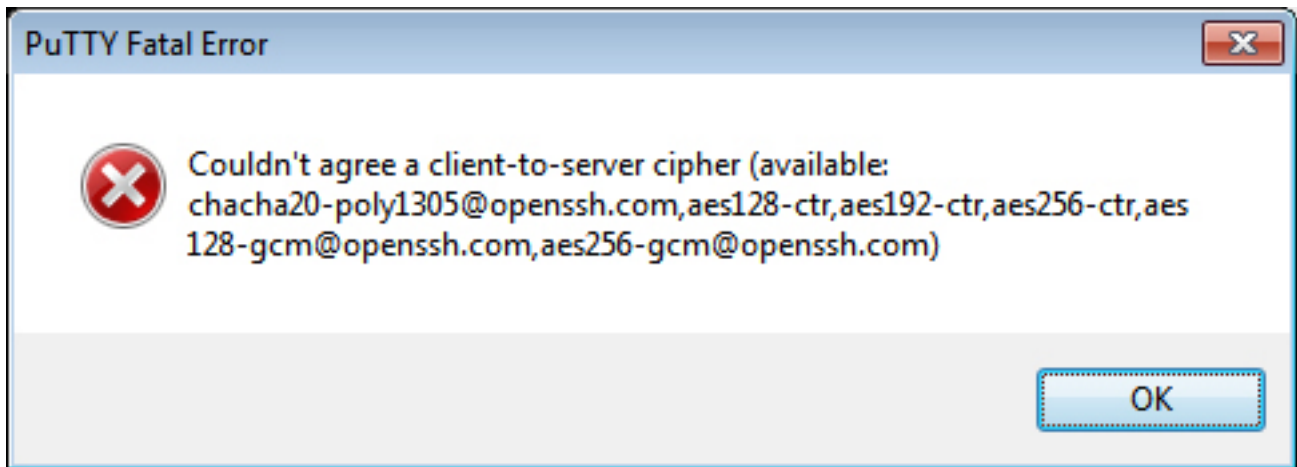
Wir machen das bei unserer ersten SSH-Verbindung jetzt auch. Wir haben die IP-Adresse herausgefunden und starten nun "putty.exe" auf unserem Windows-Rechner oder das passende Äquivalent auf einem anderen Betriebssystem.

Bei Putty schreiben wir in das Feld "Host Name (or IP address)" die IP-Adresse unseres Raspberry Pi hinein (Siehe Abbildung).



Anschließend klicken wir dann auf den Button "Open" um eine Verbindung zu unserem Raspi herzustellen. Im besten Fall fragt uns der Raspberry Pi nun nach Benutzernamen und Passwort und wir wären fertig.

Bei älteren Putty-Versionen könnte jedoch die folgende Fehlermeldung angezeigt werden:



In diesem Fall suchen wir uns eine aktuelle Version im Internet oder wir müssen noch ein bisschen konfigurieren. Dafür gehen wir zurück an unseren Raspberry Pi, öffnen ein Terminalfenster und öffnen zunächst die Datei `/etc/ssh/ssh_config`. Dazu tippen wir folgendes Kommando:

```
pi@rechnername: ~ $ sudo nano /etc/ssh/ssh_config (enter)
```

Dort treffen wir auf eine Programmzeile, die auskommentiert ist und wie folgt aussieht:

```
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc
```

Am Anfang dieser Zeile entfernt Ihr das #-Zeichen. Dann speichert Ihr mit CTRL+o (enter) und beendet den Editor mit CTRL+x.

Dieses war der Erste Streich und der zweite folgt sogleich...

Jetzt bearbeiten wir noch die Datei `/etc/ssh/sshd_config` mit folgender Kommandozeile:

```
pi@rechnername: ~ $ sudo nano /etc/ssh/sshd_config (enter)
```

An das Ende dieser Datei hängen wir folgende Zeile an (die müssen wir leider von Hand schreiben):

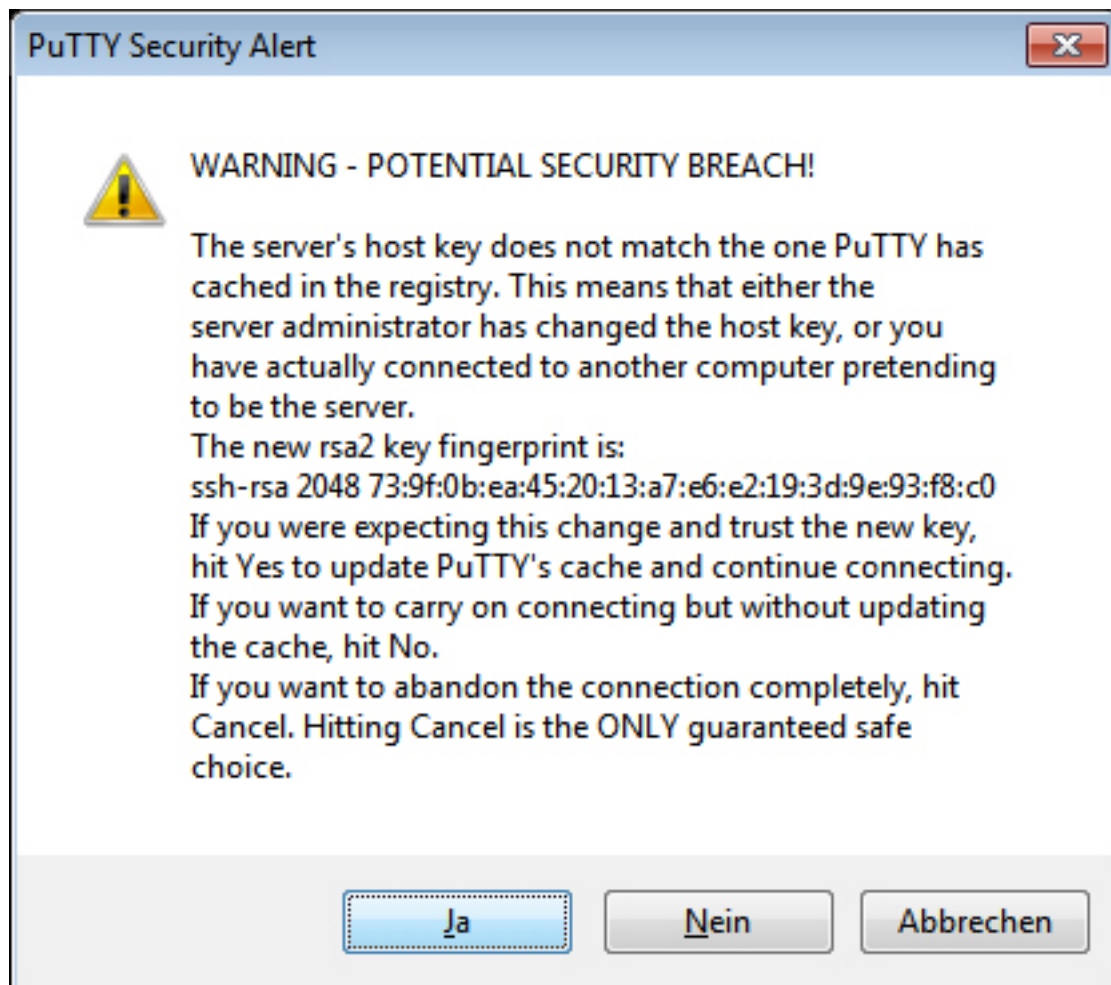
```
ciphers 3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr
```

Danach wieder CTRL+o (enter) zum Speichern und CTRL+x zum Beenden und fertig.  
Jetzt noch einmal booten:

```
pi@rechnername: ~ $ sudo reboot (enter)
```

...und dann sollten wir auch eine SSH-Verbindung zu unserem Raspberry Pi herstellen können.

Bei der ersten Verbindung beschwert sich Putty noch, dass das Zertifikat, mit dem wir uns verbinden wollen, ein potenziell unsicheres Zertifikat ist. Das können wir jedoch ignorieren, da wir ja wissen, dass unser Pi sich selbst ein Zertifikat erstellt und das natürlich auch von keiner regulären Stelle bestätigt werden kann.



Hier klicken wir einfach und völlig sorgenfrei auf den Button "Ja" und schon können wir uns mit Benutzernamen und Passwort bei unserem Raspberry Pi anmelden.

Hintenan noch ein Hinweis:

Wenn Euch der Raspberry Pi von vornherein und direkt ablehnt, obwohl im raspi-config das ssh aktiviert ist, dann versucht mal noch folgendes:

```
pi@rechnername : ~ $ sudo rm /etc/ssh/ssh_host* (enter)
pi@rechnername : ~ $ sudo dpkg-reconfigure openssh-server (enter)
pi@rechnername : ~ $ sudo reboot (enter)
```

Raspberry Pi - XRDP installieren

XRDP ist die Linux-Entsprechung des oft zur Fernsteuerung verwendeten RDP (Remote-Desktop-Protocol). Bei der Installation des Paketes wird der XRDP Server installiert. Das bedeutet, dass hier der Raspberry Pi vom Windows-PC oder natürlich auch jedem anderen RDP-Client aus ferngesteuert werden kann.

Da sich VNC und XRDP oft gegenseitig beissen, sollten wir zunächst einmal VNC deinstallieren. Das geht so:

```
pi@rechnername: ~ $ sudo apt-get purge realvnc-vnc-server (enter)
```

Danach müsst Ihr dann einfach noch das Paket XRDP installieren. Das geht so:

```
pi@rechnername: ~ $ sudo apt-get install xrdp (enter)
```

Jetzt einfach noch den Raspberry Pi neu booten und das war's:

```
pi@rechnername: ~ $ sudo reboot (enter)
```

## Backup / Datensicherung

Nachdem Ihr nun im Schweisse Eures Angesichts die Grundeinrichtung hinter Euch gebracht habt, möchte ich Euch zum Abschluss noch einen guten Rat geben.

Computersoftware, speziell Betriebssysteme sind ein extrem komplexes System an Komponenten, die im idealfall auch noch perfekt zueinander passen.

Da wir aber im realen Leben existieren, ist der Idealfall oft nicht der, mit dem wir es zu tun bekommen. Darum hier mein gut gemeinter Rat:

Der Win32 Diskimager bietet euch neben der Funktion, Images auf eine SD-Karte zu schreiben, auch die Funktion, Images von einer SD-Karte zu lesen und zu speichern.

Da es ja meistens nicht mit der Grundinstallation getan ist, man also, ggf. auch experimentell, weiter installieren und konfigurieren muss, kann hier schnell mal etwas schiefgehen.

Wenn Ihr also einen Installationsschritt abgeschlossen habt und ihr einen Softwarestand der SD-Karte erreicht habt, der sicherungswert ist, weil er mal wieder viel Zeit gekostet hat und doch nur ein Zwischenschritt ist oder wenn Ihr direkt mit einer Installation fertig seid und alles perfekt funktioniert:

**MACHT EUCH EIN IMAGE MIT DEM JEWEILIGEN SOFTWARESTAND**

Es dauert im Durchschnitt nur ein paar Minuten, eine SD-Karte wieder zurückzusichern. Eine Neueinrichtung kann wieder Stunden benötigen. Speichert Euch die Imagedateien und benennt sie mit dem Datum und beschreibt den Softwarestand im Dateinamen, dann findet Ihr sie auch wieder, wenn Ihr sie sucht.

Ausserdem lässt sich eine bestehende Grundeinrichtung natürlich auch als Basis für verschiedene Pi oder verschiedene Konfigurationen Eures einen Pi nutzen.

Ich bedanke mich für Eure Aufmerksamkeit und wünsche Euch viel Spass beim Raspeln!