Funktion Servergnomepanel in Tcos aktivieren

\+ +

Ziel ist es OpenThinClient mit einem Ubuntu 10.04 LTS nahtlos zu verbinden. Die folgenden Schritte werden hier einzeln dargestellt und erklärt. Später wird hierfür eine Vorlage erstellt, die nur mehr wenige Schritte benötigt.

Generieren eines SSH-Schlüsseln zu Authentifizierung

Dazu loggen wir uns ganz normal ein und öffnen ein **xterminal**. Jetzt lassen wir den Schlüssel bauen. Aber ohne Passphrase. Danach laden wir diesen auf unseren Server. Die Ausgaben unten bestätigen wir mit Eingabe.

A C H T U N G
Der Schlüssle muss unbedingt ohne Passphrase generiert werden, sonst ist ein späteres Einloggen nicht möglich!!!

```
ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/tcos/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tcos/.ssh/id_dsa.
Your public key has been saved in /home/tcos/.ssh/id_dsa.pub.
The key fingerprint is:
2d:a8:34:00:84:59:88:a1:76:6f:20:0d:d7:26:b4:bd ml@adl52
```

So, mit dem nächsten Befehl laden wir den soeben generierten Schlüssel auf dem Server (app) hoch. Benutzer braucht hier keiner mehr angeben werden, da diese ja identisch sind.

```
ssh-copy-id -i /home/tcos/.ssh/id_dsa.pub app
```

Als nächstes loggen wir uns am Server „app“ mit unseren soeben hoch geladenen Schlüssel ein. Dies wird aber nicht funktionieren, da dies die lokalen Sicherheitsrichtlinien des Clients nicht erlauben. Da sich das System des Clients ja bei jedem Startvorgang zusammenbaut, ist eine lokale Änderung nicht möglich. Ausgabe:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0755 for '/home/tcos/.ssh/id_dsa' are too open.
It is recommended that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: /home/tcos/.ssh/id_dsa
Enter passphrase for key '/home/tcos/.ssh/id_dsa':
ml@app's password:
```

Es wird deshalb hier wieder nach dem Passwort gefragt.

Umgehung der lokalen Sicherheitsrichtlinien

```
cp /etc/ssh/ssh_config /home/tcos/.ssh/ssh_config
vi /home/tcos/.ssh/ssh_config
```

Die Zeile `#IdentityFile … /id_dsa` so ändern:

```
IdentityFile /tmp/tcos/.ssh/id_dsa
```

Wir erstellen nun ein Script das sämtliche Aufgaben beim Einloggen automatisch ausführt. Scripte und Dateien die am Thinclient benötigt werden kopiert man wenn man am Server ist nach

```
/opt/openthinclient/server/default/data/nfs/root/custom
```

und wenn man sich am Client befindet nach

```
/var/tcos/custom
```

Wir entscheiden uns für die Bearbeitung direkt am Server, da am Server **nano** als Editor verfügbar ist.

```
mkdir /opt/openthinclient/server/default/data/nfs/root/custom/scripte
nano
/opt/openthinclient/server/default/data/nfs/root/custom/scripte/start_gnome_
app.sh
```

Diese Datei erstellen wir mit folgenden Inhalt:

start_gnome_app.sh

```
#!/bin/bash
mkdir -p /tmp/tcos/.ssh/
cp /home/tcos/.ssh/id_dsa /tmp/tcos/.ssh/.
chmod 600 /tmp/tcos/.ssh/id_dsa
ssh -F /home/tcos/.ssh/ssh_config -X app gnome-panel
```

Nun noch **ausführbar** machen:

```
chmod +x
/opt/openthinclient/server/default/data/nfs/root/custom/scripte/start_gnome_
app.sh
```

Und schon kann man durch ausführen dieses Befehls das gnome-panel vom Server app in den Desktop von OpenThinClient integrieren.

Automatischer Start des Scripts

Um das Script automatisch beim Einloggen starten zu lassen gibt es mehrere Möglichkeiten. Wir verschleiern das ganze in dem wir einen Startbefehl in in unsere **~/.profile** schreiben. (wieder am Server)

```
nano /opt/openthinclient/server/default/data/nfs/home/ml/.profile
```

```
/var/tcos/custom/scripte/start_gnome_app.sh &
```

Beim nächsten Login erscheint das Gnome-Panel des Servers „app“ automatisch und tut so als gehöre es schon immer zum Desktop.

Konfigdateien für weitere User nach "skel" kopieren

Jetzt kopieren wir noch alles in unser „custom“ Verzeichnis, damit wir nachher weitere User einfacher anlegen können. Wir befinden uns wieder am TCOS-Server (tcos) in dem Homeverzeichnis von dem User den wir gerade erstellt haben. Danach kann man zu dem Howto neuen_benutzer_in_tcos_anlegen gehen, und weitere User anlegen.

```
mkdir ../../root/custom/skel
cp .profile ../../root/custom/skel/
mkdir ../../root/custom/skel/.ssh
cp .ssh/known_hosts ../../root/custom/skel/.ssh/
cp .ssh/ssh_config ../../root/custom/skel/.ssh/
```

Script für den Userkonfigurationsassistenten

Wir erstellen eine Datei am Server die für uns dann die gesamte Erstkonfiguration eines neu angelegten Users durchführt.

```
nano /opt/openthinclient/server/default/data/nfs/root/custom/scripte/
```

Die folgenden Zeilen kopiert man einfach in die Datei und Speichert sie ab.

userinstall.sh

```
#!/bin/bash
echo ------------------------------------------------------
echo Der folgende Assistent führ Sie durch
echo die Benutzerkonfiguration
echo ------------------------------------------------------
echo
echo Zum Starten drücken Sie ENTER.
echo Zum Abbrechen drücken sie STRG+C
read x
echo Userbezogene Daten werden von "SKEL" kopiert
cp -r /var/tcos/custom/skel/.* /home/tcos/
cp -r /var/tcos/custom/skel/* /home/tcos/
echo ...fertig
echo
echo Es wird nun ein SSH-Schlüssel erstellt
echo Bestätigen Sie alle Aufforderungen mit ENTER
read x
ssh-keygen -t dsa
echo ...fertig
echo
echo Der generierte Schlüssel wird nun auf den Server
echo app kopiert, geben Sie bei Aufforderung Ihr
echo persönliches Passwort ein. Weiter mit ENTER
read x
ssh-copy-id -i /home/tcos/.ssh/id_dsa.pub app
echo ...fertig
echo
apt-get moo
loops=10
echo "Installationsprozess:"
step=$(( loops / 10 ))
div=$(( loops / 10 ))
prog=""
for i in `seq $loops`; do
test $(( i % step )) -eq 0 && prog="$prog#"
printf "\e[G1\e[2KProgress: % -10s %.2f" "$prog" $(( i / div ))
i=$(( i + 1 ))
sleep 1
done
echo
echo
echo "GRATULATION!!!"
echo Die Konfiguraton des Users ist jetzt abgeschlossen
echo Sie können sich neu am Client anmelden
echo "HAVE A LOT OF FUN"
echo
```

Jetzt müssen wir sie noch Ausführbar machen.

```
chmod +x
/opt/openthinclient/server/default/data/nfs/root/custom/scripte/userinstall.
sh
```

MYHOME an den Thinclient binden

Hat man nun endlich seinen User angelegt, ist eingeloggt, und Arbeitet frisch fröhlich dahin, kommt man sehr schnell drauf das die Standardverbindung über SSH mit dem Server nicht so ganz klappt. Um das ganze zu automatisieren, mit richtigen Benutzerrechten usw., muss man unbedingt für jeden User der angelegt wird auch das Homeverzeichnis vom Server „app“ mit den richtigen Rechten über eine verschlüsselte Leitung hinzumounten. Ein Beispieleintrag in der fstab könnte so aussehen:

```
sshfs#ml@app:/home/ml
/opt/openthinclient/server/default/data/nfs/home/ml/MYHOME fuse _netdev 0 0
```

Damit das ganze auch ohne Passwort funktioniert, muss man als root am Server „tcos“ einmalig einen geheimen Schlüssel generieren. Die Abfragen einfach immer mit „EINGABE“ bestätigen.

```
ssh-keygen -t dsa
```

Der Öffentliche Schlüssel von root muss aber für jeden User zum Server „app“ kopiert werden.

```
ssh-copy-id -i /root/.ssh/id_dsa.pub ml@app
```

Ob alles funktioniert hat, kann man mit einem **mount -a** überprüfen. Der Ordner ist jetzt sofort verfügbar.

```
ml@app:/home/ml on
/opt/openthinclient/server/default/data/nfs/home/ml/MYHOME type fuse.sshfs
(rw,max_read=65536)
```

Weitere Informationen

- http://www.canonical.com
- http://www.ubuntu.com
- Ubuntuusers Wiki SSH

From:
https://deepdoc.at/dokuwiki/ - **DEEPDOC.AT - enjoy your brain**

Permanent link:
https://deepdoc.at/dokuwiki/doku.php?id=prebuilt_systems:open_thinclient:funktion_servergnomepanel_in_tcos_aktivieren

Last update: **2017/04/03 21:05**