it-versity

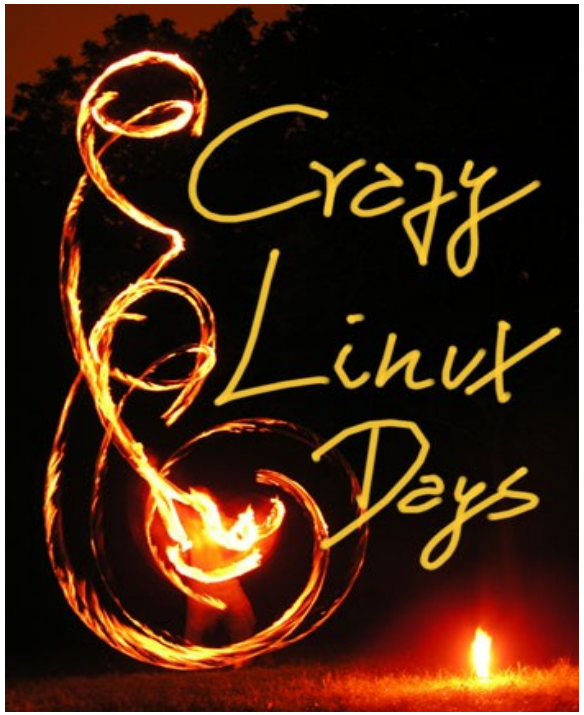
Wolfgang Ziegler und Partner Ges.m.b.H.

Schottenfeldgasse 69

A – 1070 Vienna

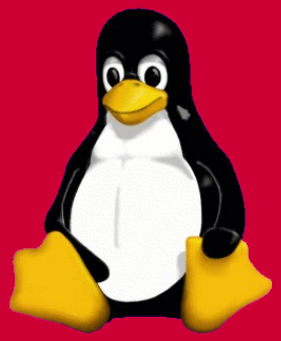
Phone +43 (1) 5228222

servicecenter@it-versity.com



Linux: Secure Shell - SSH

- OpenSSH-Grundlagen
- Server & Client Konfiguration
- Anmeldevarianten
- Einbinden in das Dateisystem



SSH Grundlagen

Grundlagen

- Symmetrische Verschlüsselung
  - DES (56bit)
  - 3DES (112 oder 168bit)
  - IDEA (128bit)
  - Blowfish (448bit)
  - AES (128, 192, 256bit)
- Asymetrische Verschlüsselung
  - RSA
  - DSA
  - Diffie-Hellman

SSH Funktionen

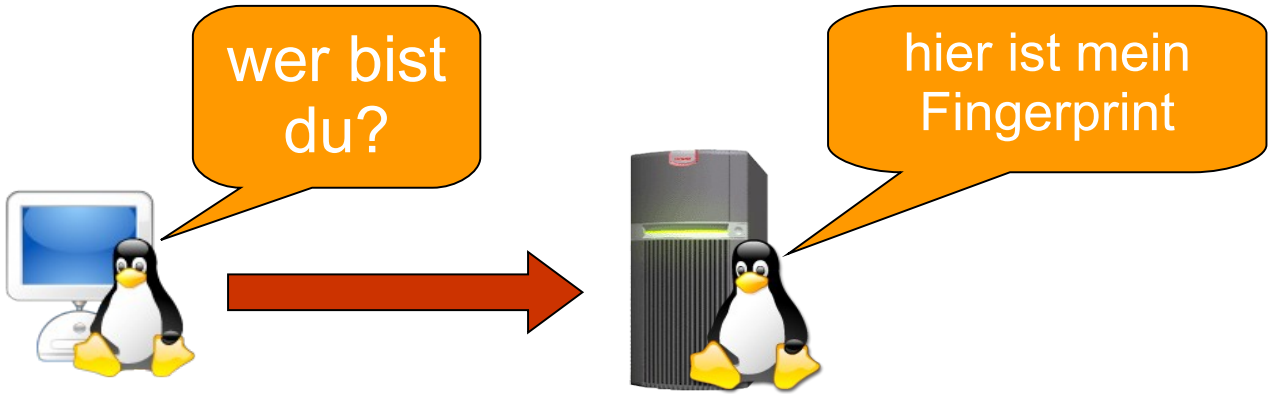
- Vorteile von Secure Shell
  - Interactive Ausführung von Befehlen
  - Sicheres kopieren von Dateien zwischen den Hosts
  - sehr schnelle sichere Authentifizierung
  - Verschlüsselte Datenübertragung
  - Tunneling und Port Forwarding
  - Ersatz für rlogin, rsh and rcp
  - Verfügbar auf vielen Plattformen
  - ausführen von SSH-Loginscripts

SSH Authentifizierung

- Details zur SSH Authentifizierung
  - sehr schnelle, sichere Authentifizierung
  - Jede SSH Verbindung verlangt zwei Authentifizierungen (Server & Client)
  - verschiedene Auth. Varianten sind verfügbar
  - Mögliche Zugriffskontrolle durch:
    - DenyUsers
    - AllowUsers
    - DenyGroups
    - AllowGroups

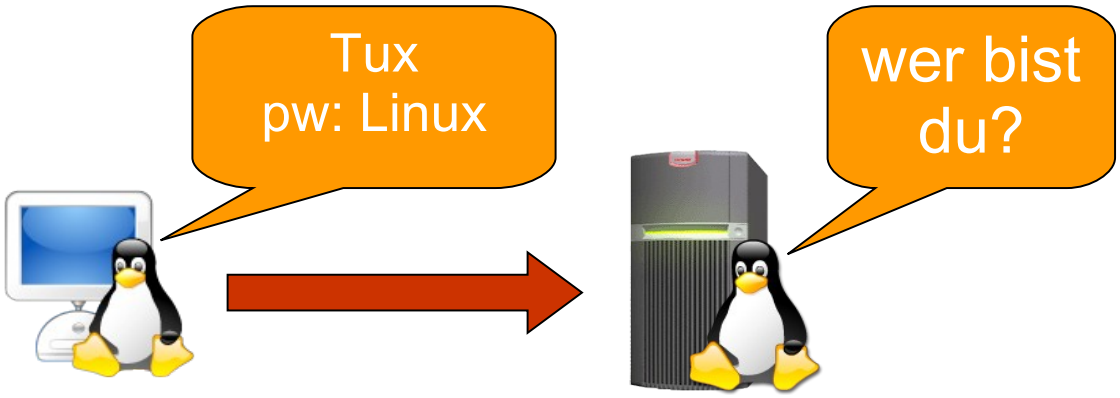
Server Authentifizierung

- Server Authentifizierung
  - der Client überprüft die Identität des Servers
  - schützt vor Man-in-the-Middle-Angriffen



Client Authentifizierung

- Client Authentifizierung
  - der Server überprüft die Identität des Client
  - Login mit Benutzername und Passwort
  - Login mit PublicKey
  - ...



- Secure Shell Version 1

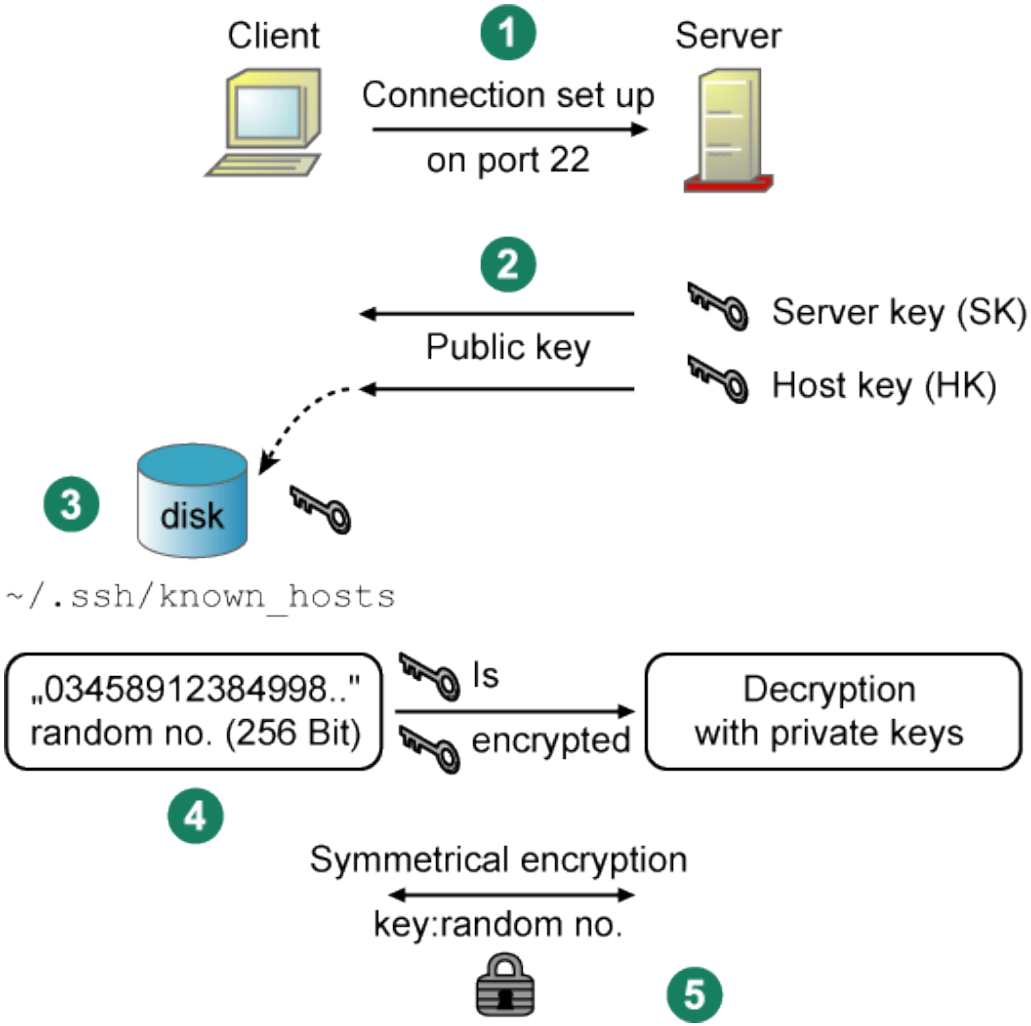
  beinhaltet keinen Mechanismus um nicht in die Verbindung Datenpakete ein zu schmuggeln (insertion attack)

- Secure Shell Version 2

  wurde um HMAC und Diffie-Hellmann erweitert

Mit telnet auf Port 22 kann überprüft werden ob der Server Version 1 oder 2 bietet

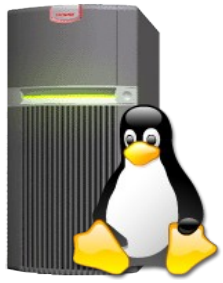
Verbindungsaufbau - SSH2



Verbindungsaufbau - SSH2

SSH-CLIENT

1. Verbindungsanfrage →
2. ← Hostauthentifizierung
3. Sitzungsschlüssel festlegen ↔
4. Benutzerauthentifizierung →
5. ← Start der Datenübertragung



SSH-SERVER

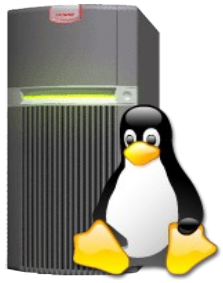
SSH Verbindungsaufbau -1-

SSH-CLIENT

1

Verbindungsanfrage

auf Port 22



SSH-SERVER

- Der Client sendet eine Verbindungsanfrage an den Port 22 des entfernten Rechners

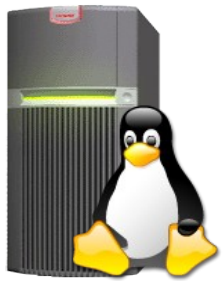
- Sollte am Server eine Firewall aktiv sein, muß der Port 22 geöffnet werden

SSH Verbindungsaufbau -2-

SSH-CLIENT

2

Hostauthentifizierung



SSH-SERVER

- Der Server authentifiziert sich mittels seines Hostkeys der bei der Installation erstellt wurde
- Der Client überprüft ob ihm der Hostkey bereits bekannt ist. Wenn nicht fragt er den Benutzer ob er den Key akzeptieren soll und speichert ihn in

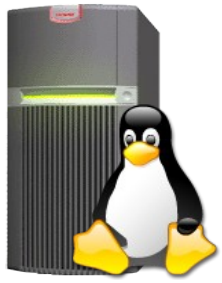
  ~/.ssh/known_hosts

SSH Verbindungsaufbau -3-

SSH-CLIENT

3

Sitzungsschlüssel festlegen



SSH-SERVER

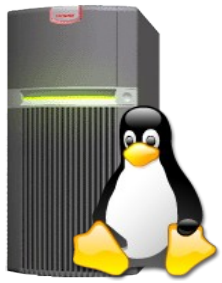
- Der Client und der Server handeln sich einen symmetrischen Sessionkey aus
- Ab jetzt wird die restliche Kommunikation mit dem Sessionkey verschlüsselt

SSH Verbindungsaufbau -4-

SSH-CLIENT

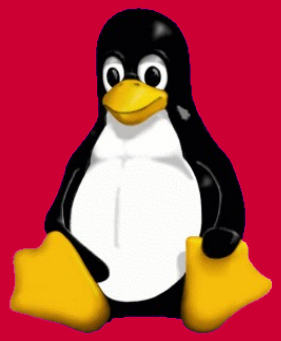
4

Benutzerauthentifizierung



SSH-SERVER

- Nun authentifiziert sich der Benutzer am Server
- Authentifizierung kann erfolgen über:
  - Benutzername und Kennwort
  - Public-key Authentifizierung
  - Kombination aus beiden Punkten



Konfigurationsdateien

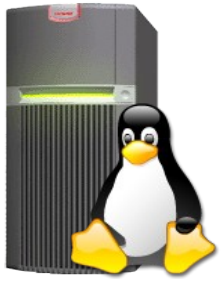
Konfigurationsdateien

- SSH Server Konfigurationsdatei

/etc/ssh/sshd_config ...SSH-Server

/etc/ssh_known_hosts ...vertrauenswürdige Hosts

~./ssh/authorized_keys ...Zertifikatsspeicher



SSH-SERVER

Konfigurationsdateien

- SSH Client Konfigurationsdatei

/etc/ssh/ssh_config	...Allgemeine Client Einstellung
~/.ssh/config	...User spezifische Einstellung
~/.ssh/known_hosts	...vertrauenswürdige Hosts
~/.ssh/id_rsa	...Privater RSA Key
~/.ssh/id_rsa.pub	...Öffentlicher RSA Key

SSH-CLIENT

Vorrangregel

- Abarbeitung der Konfigurationsdateien
  - Optionen beim SSH-Aufruf
  - ~/.ssh/config
  - /etc/ssh/ssh_config

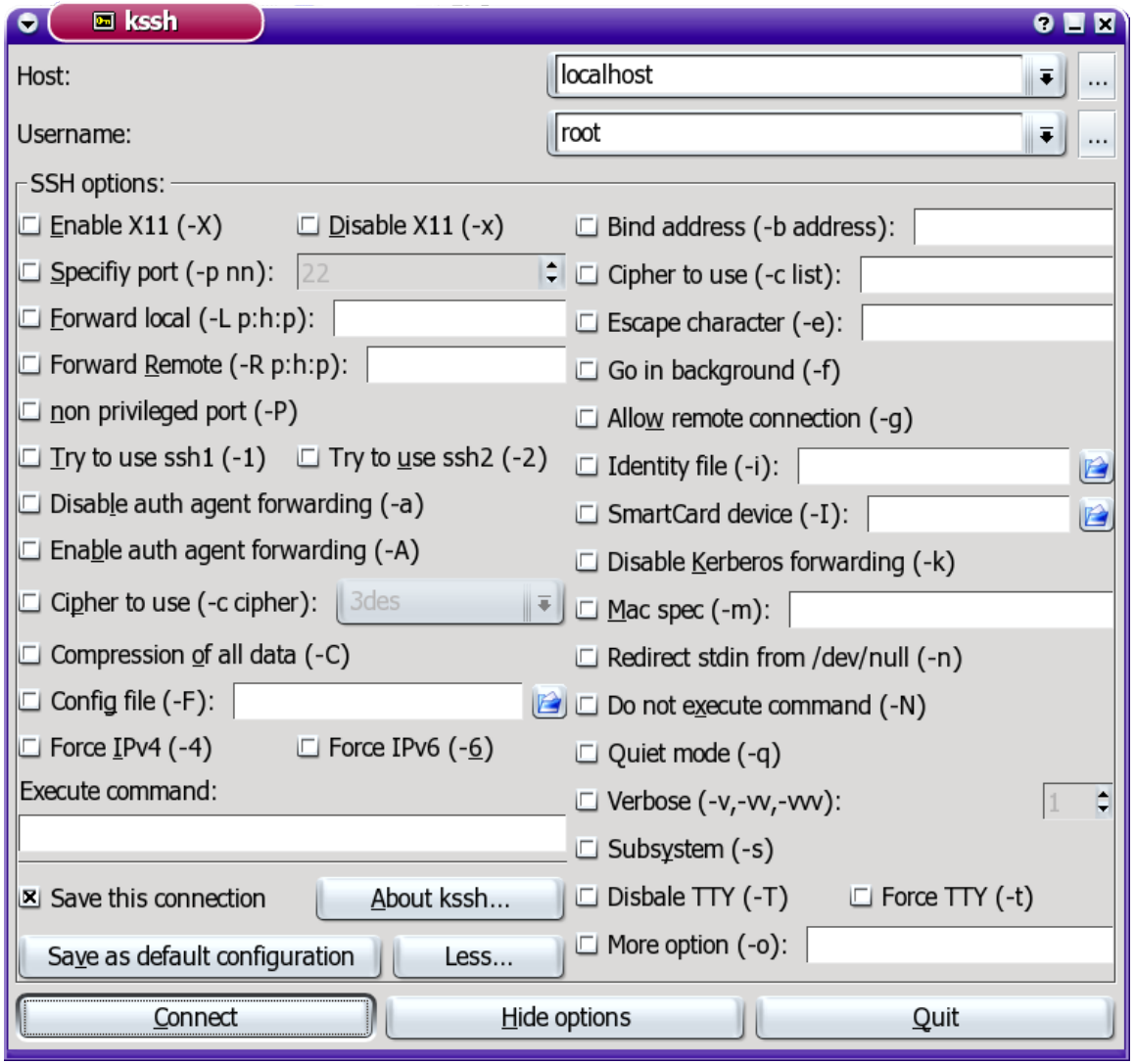
- Sonderfall SSH:
  - Optionen aus Schritt 1-3 werden summiert
  - Gesetzte Optionen werden durch später gesetzte Optionen nicht mehr überschrieben!

Hostabschnitte

- Einteilen der Client Konfig. in Host Abschnitte
  - die Optionen können pro Host gruppiert werden
  - Abschnitt endet bei nächstem Host Eintrag

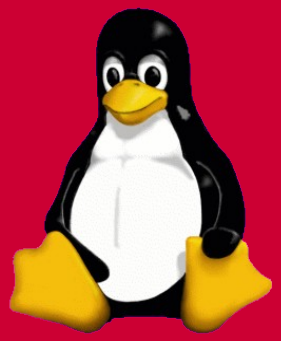
```
Host  server1
    Optionen
    Optionen
Host  server1
    Optionen
Host  *
    Optionen für restliche Rechner
```

SSH-Konfiguration mit KSSH



SSH Client Befehle

- Verbindungsaufbau
  - ssh ...SSH Client
  - scp ...sicheres kopieren von Dateien
  - sftp ...wie ftp jedoch verschlüsselt
- Erstellen & Verwalten von Keys
  - ssh-add ...registriert neue Keys
  - ssh-agent ...Verwaltet private RSA-Keys
  - ssh-keygen ...erstellt neue RSA-Keys



Anwendungsbeispiele

SSH Client

- Verbindungsaufbau mit aktuellen Benutzer

  ssh asterix.triples.at

  ssh -X asterix.triples.at ...inkl. X11

- Verbindungsaufbau mit Benutzerangabe

  ssh -l tux asterix.triples.at

  ssh tux@asterix.triples.at

X-Window und SSH

- Vorraussetzung für SSH mit X-Window
  - In der Server Konfiguration erlaubt
  - SSH-Client mit -X gestartet oder generell in der Client Konfiguration voreingestellt

/etc/ssh/ssh_config

ForwardX11 yes

Compression yes

/etc/ssh/sshd_config

X11Forwarding yes

einzelne Befehle ausführen

SSH CLIENT OPTIONEN	
-t	Prozess wird im Vordergrund gestartet
-f	Prozess wird nach Anmeldung im Hintergrund gestartet

- Starten eines einzelnen Befehls über ssh

```
ssh -t  root@server2   df -h

ssh -fX  root@172.16.0.1   yast2
```

SSH Dateimanager Integration

- KDE-Konqueror

  fish://peter@rechnername

- GNOME-Nautilus

  ssh://root@rechnername

TIPP

- SSH-Programme für Windows
  - Putty (SSH Client)

    http://www.putty.nl/download.html
  - WinSCP (SCP-Client)

    http://winscp.net/eng

- SSH-Programme für MAC
  - Fugu (SSH/SFTP Client)

    http://rsug.itd.umich.edu/software/fugu

SSH Optionen

it-versity

```
ssh -optionen Zielserver
```

SSH CLIENT OPTIONEN	
-X (groß)	mit Zugriff auf grafische Programme (X-Forwarding)
-x (klein)	ohne Zugriff auf grafische Programme
-C	Verwendung der Datenkompression
-L	Lokalen Port an Server weiterleiten
-R	Remote Port an Client weiterleiten
-l	Benutzername
-2	unterstütze nur SSH Version 2

Sicheres Kopieren

- Vorteile von scp (secure copy)
  - verschlüsselte Authentifizierung
  - Daten werden verschlüsselt übertragen
  - nach dem kopieren wird die Verbindung abgebaut
  - Bandbreitenmanagement

```
 # Kopieren einer Datei mit scp
peter@asterix:~> scp /etc/fstab  root@172.16.0.120:/tmp/fstab.asterix
Password:
fstab                                         100%  711     0.7KB/s   00:00
peter@asterix:~>
```

Sicheres Kopieren

```
scp  /etc/fstab  asterix.triples.at:
```

Quelle

Ziel

- Datei auf einen anderen Rechner kopieren

  scp /etc/motd asterix.triples.at:

- Datei von einem anderen Rechner kopieren

  scp tux@asterix.triples.at:/etc/hosts /tmp

SCP Optionen

it-versity

```
scp -optionen Quelle Ziel:Zielort
```

SCP OPTIONEN	
-2	unterstütze nur SSH Version 2
-l xxx	Bandbreite in KBit/s
-C	Verwendung der Datenkompression
-P	Zu verwendender Port
-p	Rechte & Zeitstempel beim kopieren beibehalten
-r	Rekursives kopieren inkl. aller Verzeichnisse
-v	Verbose Modus

SCP Beispiele

- Kopieren eines Verzeichnisses

  scp -r /etc 172.16.0.131:

- Kopieren an einen speziellen Ort

  scp -r /etc 172.16.0.131:/tmp

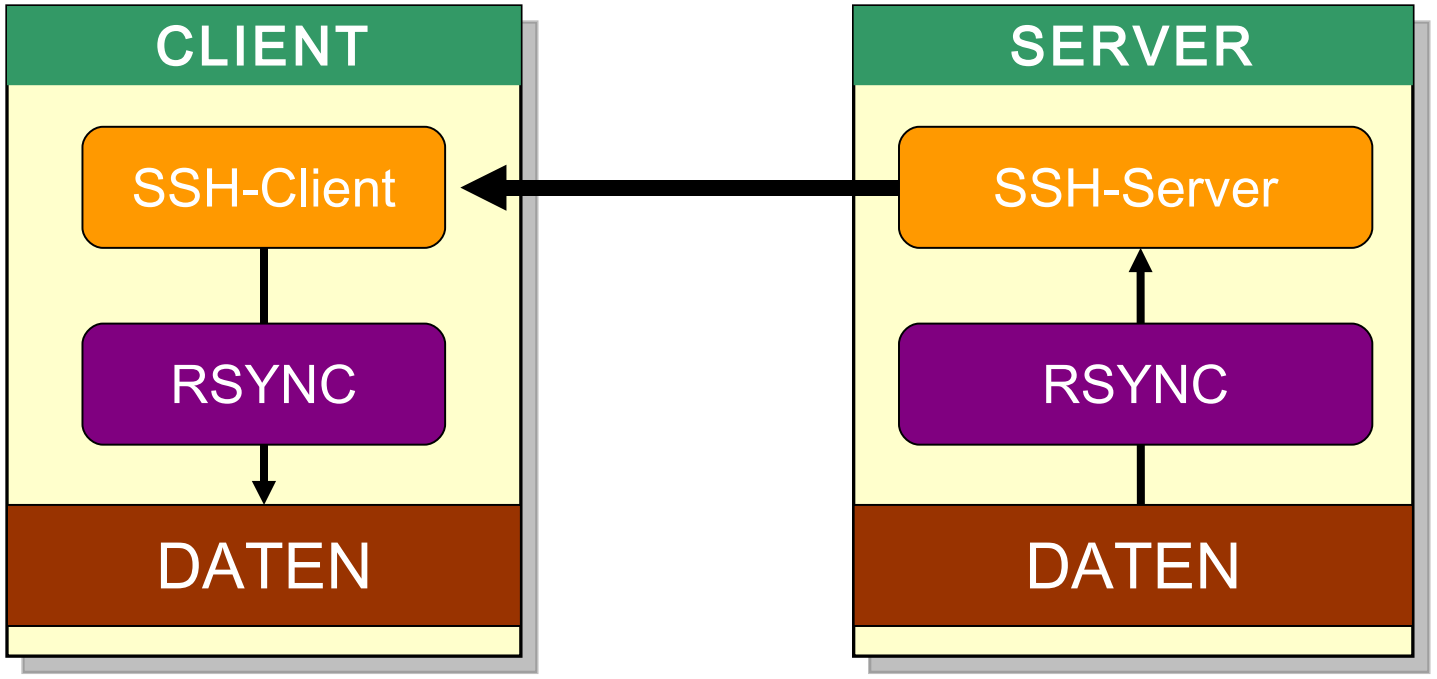
- Kopieren mit Bandbreitenmanagement

  scp -r -C -l 300 /etc 172.16.0.131:/tmp

Sicheres Backup

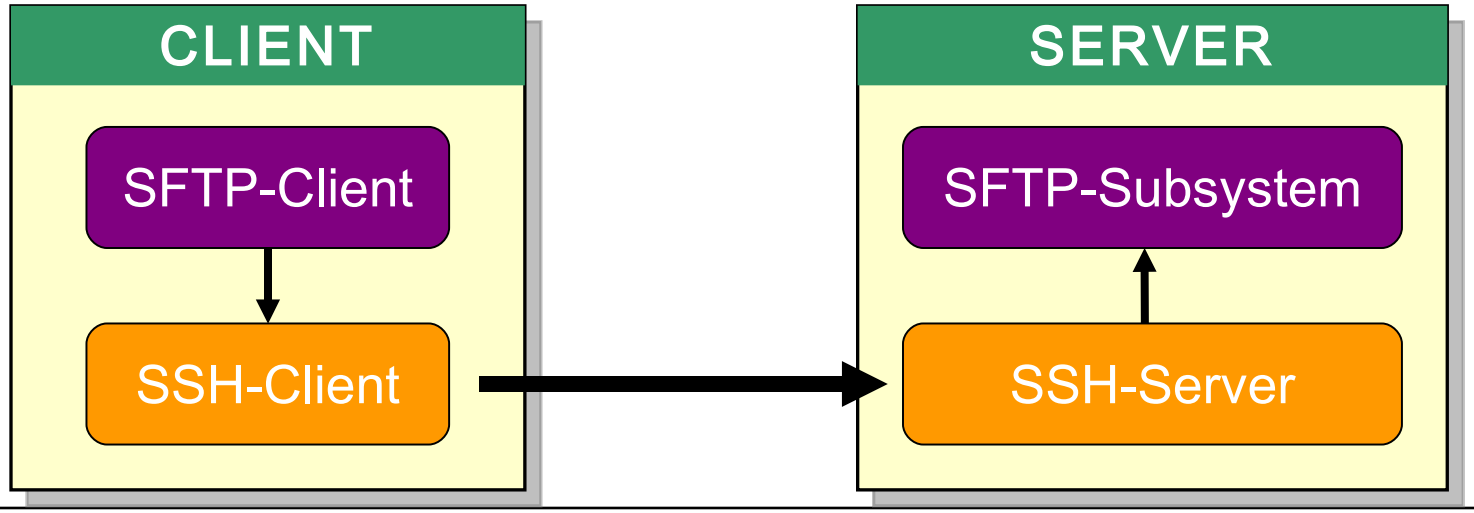
- Sichern eines Remote Servers mit rsync

  rsync -ave ssh root@server:/home/geeko /backup/



SSH File Transfer Protocol

- Vorteile von sftp (SSH-FTP)
  - Bedienung wie bei einem FTP-Client
  - kann nicht mit einem FTP Server verwendet werden
  - verschlüsselte Authentifizierung durch SSH
  - Daten werden durch SSH verschlüsselt übertragen

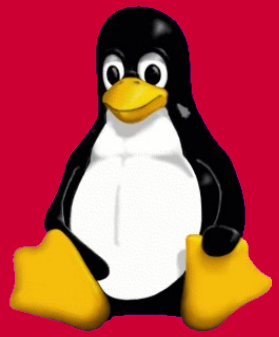


Sicheres Kopieren

- Verbindungsaufbau über sftp

  sftp jahn@asterix.triples.at

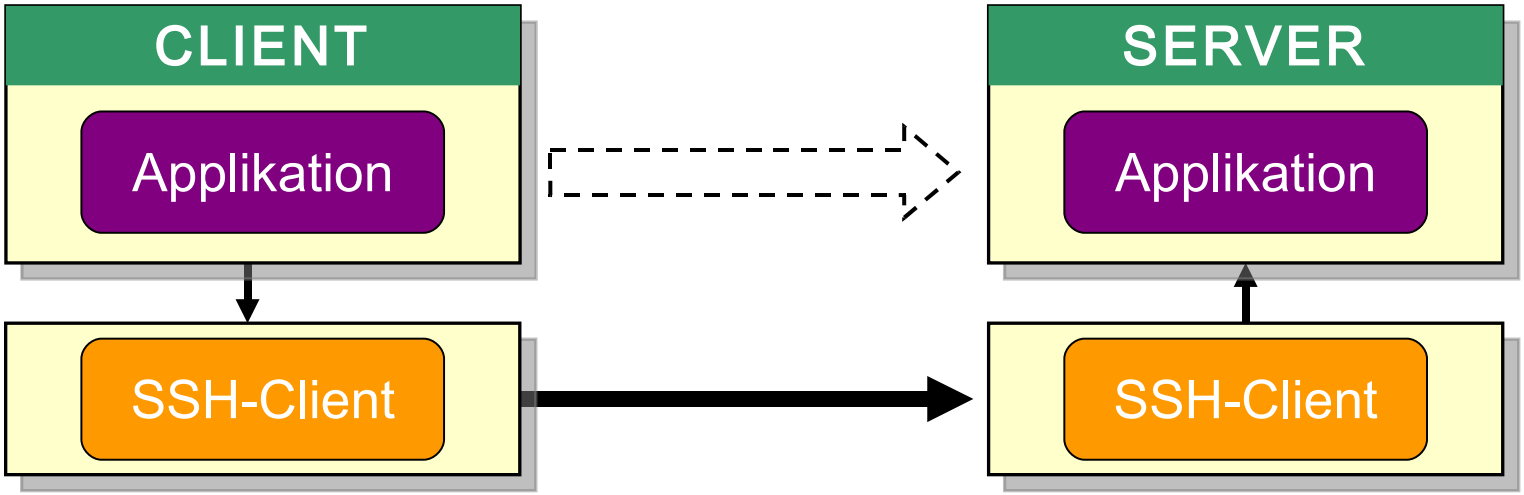
```
 # Verbindungsaufbau über  SFTP
peter@asterix:~> sftp root@172.16.0.120
Connecting to 172.16.0.120...
The authenticity of host '172.16.0.120 (172.16.0.120)' can't be established.
RSA key fingerprint is ee:45:4f:5b:54:4e:2f:cf:de:62:d4:21:4b:65:ad:42.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.0.120' (RSA) to the list of known hosts.
Password:
sftp>
```



Forwarding

Port Weiterleitung

- Port Weiterleitung (Port Forwarding)
  - Die Kommunikation einer unsicheren Anwendung verläuft durch einen sicherenSSH-Tunnel
  - ist nur für TCP-basierte Protokolle möglich
  - z.B POP3, SMTP, HTTP, FTP,...



Arten der Port Weiterleitung

- Lokales Forwarding
  - TCP-Client (Applikation) und SSH-Client befinden sich auf dem selben Host
  - lokalen Port für Kommunikation umbiegen
- Remote Forwarding
  - TCP-Client (Applikation) und SSH-Server befinden sich auf dem selben Host
  - remote Port für Kommunikation umbiegen

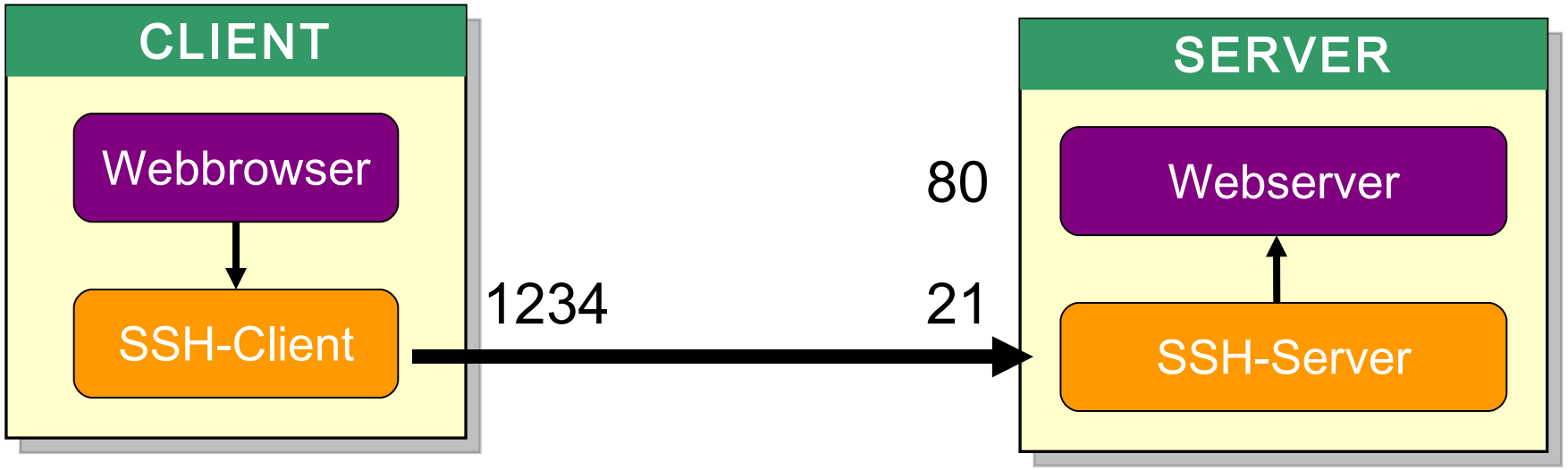
Priveligierte Ports 0-1024 können nur von Root weitergeleitet werden

Lokales Forwarding

- Port 1234 auf Remote Host port 80 tunneln

  ssh -L 1234:localhost:80 server

  Testen vom Client: http://localhost:1234

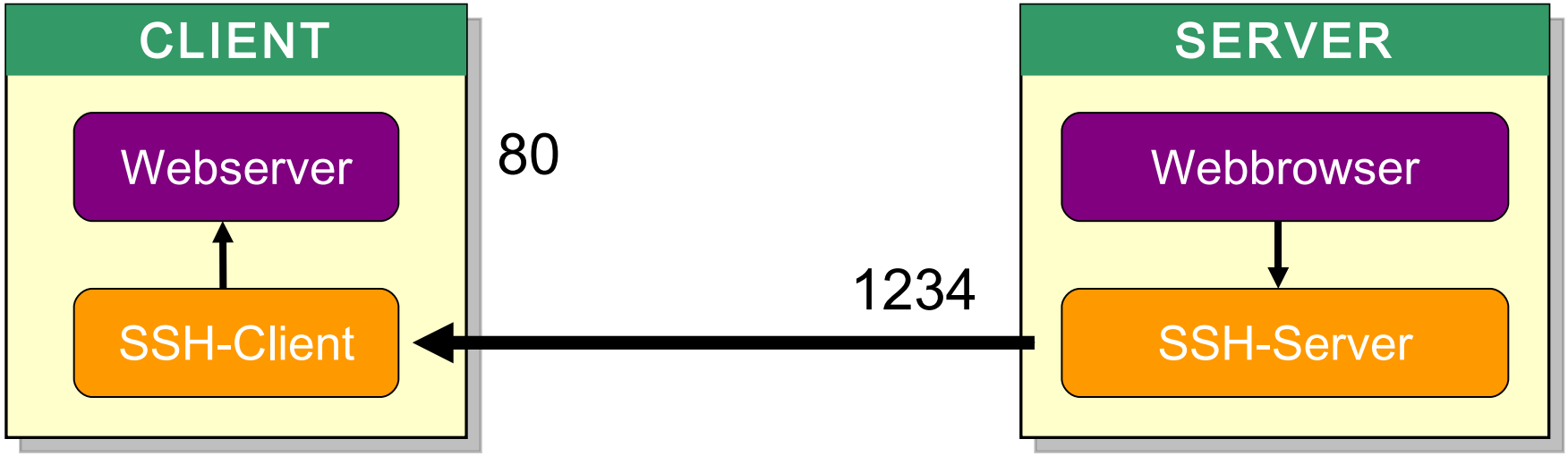


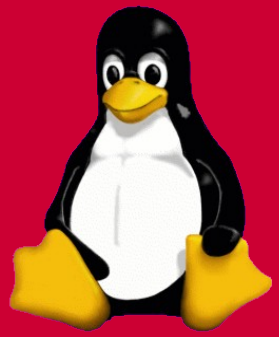
Remote Forwarding

- Port 1234 auf Lokalen Host port 80 tunneln

  ssh -R 1234:localhost:80 server

  Testen vom Server: http://localhost:1234





Zugriffskontrolle

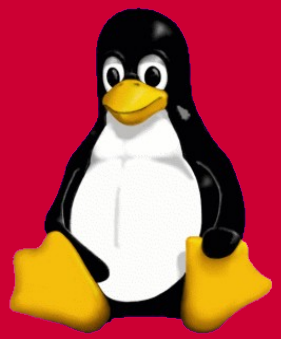
Zugriffskontrolle

- Sperren von einzelnen Benutzer Accounts
  1. DenyUsers
  2. AllowUsers
  3. DenyGroup
  4. AllowGroup

- Reihenfolge der Auswertung
  - 1 hat eine größere Priorität als 2 oder 4

Zugriffskontrolle

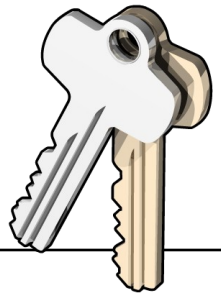
- Beispiele für die Zugriffskontrolle
  - DenyUsers root@*.triples.at gustav
  - AllowUsers *@*.itversity.at
  - DenyGroups users vertrieb
  - AllowGroups vert* abteilung?



einfache Public-Key Authentifizierung

Public-Key Authentifizierung

- Einrichten der Public-Key Authentifizierung
  - Erzeugen von SSH-2 Key Paaren am Client
  - Kopieren des Client public Key auf den Server
  - Importieren des public Key in authorized_keys



Public-Key Authentifizierung

it-versity

- Erzeugen von SSH-2 Key Paaren

  ssh-keygen -t rsa

  ~/.ssh/id_rsa ...private Key

  ~/.ssh/id_rsa.pub ...public Key

```
 # Erzeugen der Key Paare
peter@asterix:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/peter/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/peter/.ssh/id_rsa.
Your public key has been saved in /home/peter/.ssh/id_rsa.pub.
The key fingerprint is: ce:d8:...:00:de:67:6b:c1:d6 peter@asterix
```

Public-Key Authentifizierung

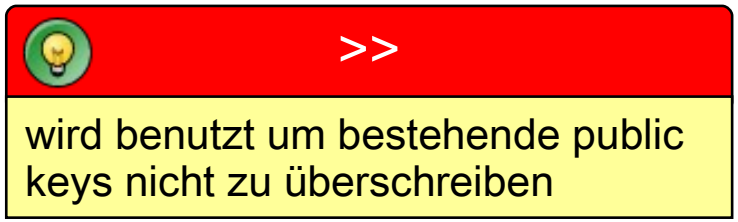
- Kopieren des public Key auf den Server

  scp .ssh/id_rsa.pub server:asterix.pub

- Importieren des public Key

  ssh server

  cat asterix.pub >> .ssh/authorized_keys



SSH-Login

- PasswordAuthentication yes

  Login mit Benutzername & Passwort ist erlaubt

- PubkeyAuthentication yes

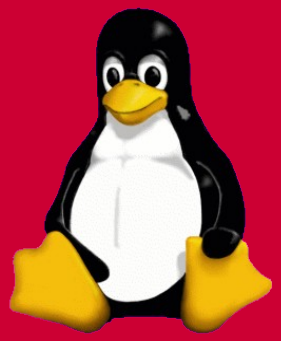
  Login mit public key authentication ist erlaubt

  (SSH-Version2)

SSH Login

- Login NUR über Public Key erlauben
  - PasswordAuthentication no
  - PubkeyAuthentication yes
  - UsePAM no ...nur bei neueren SSH-Versionen

Nach Ändern der Konfigurationsdatei muss der Dämon neu gestartet werden: rcsshd restart



verschärfte Public-Key Authentifizierung

Passphrase

- Vorteile einer Passphrase
  - zusätzliche Sicherheitsstufe
  - ohne Wissen über die Passphrase ist der private Key nutzlos (Diebstahl)
- Nachteile einer Passphrase
  - Passphrase muss beim Verbindungsaufbau eingegeben werden
  - Passphrase Eingabe kann durch ssh-agent automatisch erfolgen

automatische Passphrase

- Starten des ssh-agent

  eval 'ssh-agent'

- Private Key an Agent übergeben

  ssh-add ~/.ssh/id_rsa ...Parameter ist Optional

Passphrase Übergabe funktioniert nur in der Shell in der die Befehle eingegeben wurden.

automatische Passphrase

Passphrase Übergabe funktioniert nur in der Shell in der die Befehle eingegeben wurden.

Besser fix eintragen in:

~/.profile ...gilt für alle Shells

~/.xinitrc ...gilt für die ganze X11-Sitzung

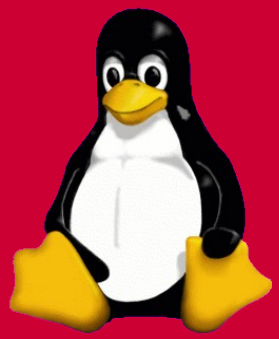
- Script Beispiele

  http://wiki.koeln.ccc.de/index.php/SSH_Agent

  http://www.rrzn.uni-hannover.de/ssh_profile.html

Verschärfte Einstellung

- StrictHostKeyChecking yes
  - nur Server welche einen Eintrag in known_hosts haben werden akzeptiert
  - neue Einträge müssen manuell in die Datei eingefügt werden



SSH Einbindung ins Dateisystem

Dateisystem Einbindung

- SSH transparent ins Dateisystem einbinden
  - Dateisystemtreiber SHFS

    http://shfs.sourceforge.net
  - Dateisystemtreiber SSHFS

    http://fuse.sourceforge.net/sshfs.html

Installation SHFS - RPM

- Erstellen eines SHFS RPM Paketes
  1. SHFS-Source Code entpacken
  2. SHFS-Patches installieren
  3. Source Code kompilieren
  4. RPM-Paket erstellen
  5. RPM-Paket installieren

Installation SHFS - RPM 1

- Erstellen eines SHFS RPM Paketes
  1. tar -xvzf shfs-0.35.tar.gz
  2. cd shfs-0.35

Patches installieren - RPM 2

- Download der SHFS-Patches von:

  http://atrey.karlin.mff.cuni.cz/~qiq/src/shfs/shfs-0.35/

  d_entry-2.6.16.diff ...Kernelpatch ab 2.6.16

  gcc4-compilefix.patch ... GCC Patch ab Version 4

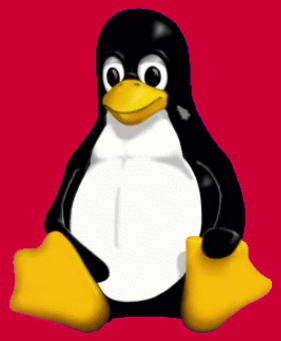
- Installation der Patches
  - cd shfs-0.35
  - patch -p0 < d_entry-2.6.16.diff
  - patch -p1 < gcc4-compilefix.patch

Installation SHFS - RPM 3

- Erstellen eines SHFS RPM Paketes
  - make
  - chown -R root.root rpm/
  - Wortlaut Copyright: GPL durch License: GPL ersetzen in folgenden Dateien

    rpm/shfs-utils.spec.in

    rpm/shfs-module.spec.in
  6. make rpm



Anwendung von SHFS

Mounten mit SHFS

- Mögliche Arten des SHFS-Mounten
  - Händisch: shfsmount
  - Automatisch: /etc/fstab
- Dateirechte & Freigaben
  - Zugriff erfolgt nur aufgrund der Dateirechte

SSH-Mount

- Mounten des Heimatverzeichnis von Peter

```
shfsmount peter@asterix  /mnt/asterix
```

- Mounten von /etc

```
shfsmount peter@asterix:/etc  /mnt/asterix
```

SHFS Optionen

SHFS OPTIONEN	
nocache	deaktiviert den Read/Write Cache
preserve	erhält die Besitzer- und Gruppeninformationen des Servers auf dem Client
uid=Nutzer, gid=Gruppe	legen den Besitzer der Dateien auf dem Client fest
rmode=xxx	legt die Rechte des Mountpoints fest
persistent	baut die Verbindung erneut auf, falls sie getrennt wird
...	

Mappen der Dateirechte I

- Benutzer existiert auf beiden Systemen
  - beim mounten eine Verbindung zwischen den Benutzer herstellen

```
shfsmount peter@asterix.triples.at /mnt/asterix  \
-o preserve,rmode=755
```

Mappen der Dateirechte II

- Benutzer ist unterschiedlich auf beiden Systemen
  - beim mounten eine Verbindung zu einem anderen Benutzer herstellen

```
shfsmount peter@asterix.triples.at /mnt/asterix  \
-o uid=jahn,gid=users
```

jahn ...ist der lokale Benutzer

SHFS-Root Login

- Root Login über SHFS

muss in /etc/ssh/sshd_config aktiviert werden

PermitRootLogin yes

- Benutzern das SHFS-Mounten erlauben
  - Verzeichnis für Mountpoint muß den Benutzer gehören
  - Benutzer muss Rechte auf den Mount Befehl haben

```
chmod u+s $(which shfsmount)
chmod u+s $(which shfsumount)
```

Crazy Linux Days