Vorlesung Unix Praktikum

TechFak-Account und Netboot

Dorian Lehmenkühler

dorian@techfak.de

Rechnerbetriebsgruppe
Technische Fakultät
Universität Bielefeld

16. Oktober 2019

Was steht heute an?

1. Remote arbeiten allgemein
   - SSH
   - Compute
   - Mehr SSH

2. Remote arbeiten von unterwegs
   - Einloggen via shell
   - Dateien übertragen via files

Remote arbeiten
SSH

ssh = **s**ecure **sh**ell („Sichere Kommandozeile“)

- Ermöglicht Login auf entfernte Rechner
- Gibt einem eine Kommandozeile

Befehl:
`$ ssh [USER@]HOST`

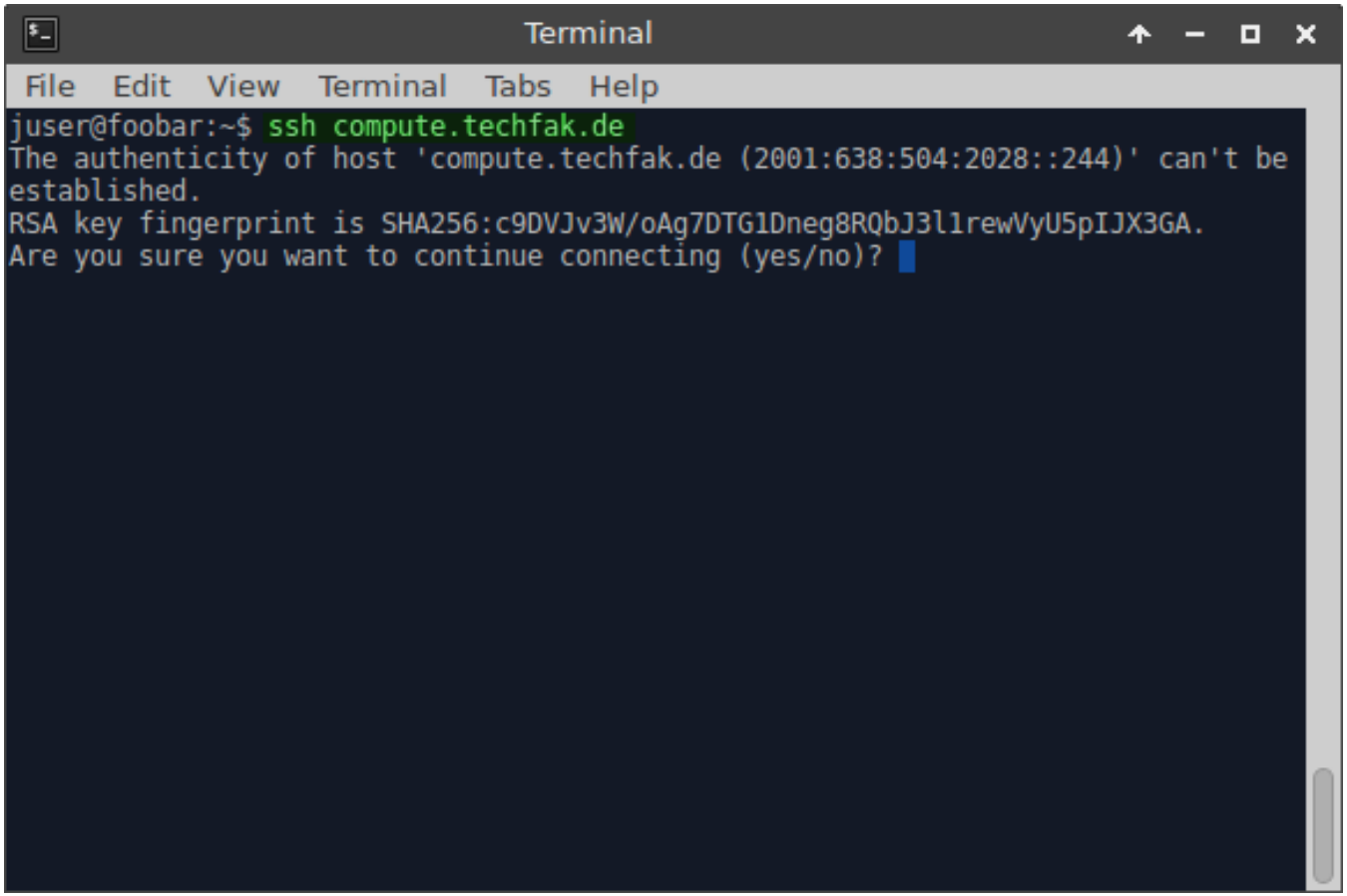
Beim ersten Verbinden zu einem Rechner wird der Fingerprint des Hosts angezeigt.
→ vergleichen, um Authentizität des Hosts zu bestätigen!

Angreifer könnte z.B. mit compute.techfka.de PWs abgreifen

Remote arbeiten

SSH

Beispiel: compute.techfak.de



Remote arbeiten

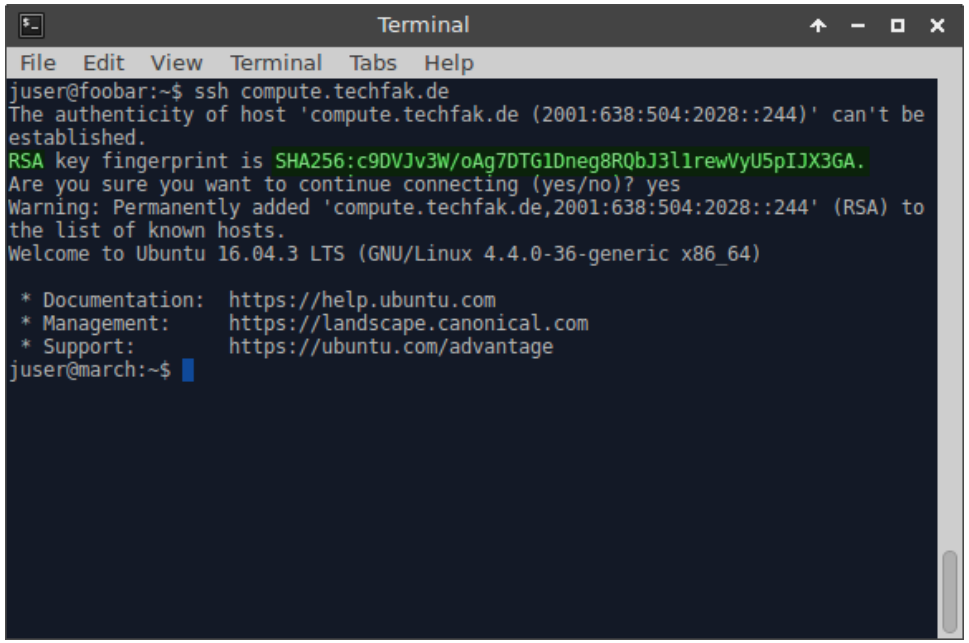
SSH

Auf `techfak.net/compute` nachsehen:

Die aktuellen SSH-Fingerprints sind

SHA256:

```
/9g6ihqzEGBGFJmC/xT5QV8L9wfCjCdwb+hTTKWinEw (ED25519)
mX9jq6gw0JueY8O+zEC0hIRmdqpn8rlux6VUtqpbIp8(ECDSA)
c9DVJv3W/oAg7DTG1Dneg8RQbJ3l1rewVyU5pIJX3GA (RSA)
```



Remote arbeiten

Die Compute Maschinen

- compute.techfak.de
- Mehrere identische Netboot-Maschinen mit viel Rechenleistung
- Immer an
- Aus dem gesamten TechFak-Netz per ssh erreichbar

- Verbinden immer mit compute.techfak.de

→ Einzige Ausnahme: Job läuft schon auf bestimmter Maschine

Remote arbeiten

Die Compute Maschinen

- Lang laufende Jobs können im `tmux` gestartet werden
- Rechenintensive/lange Prozesse bitte mit `nice` starten

`$ tmux`	Tmux starten
`$ nice meinprozess.sh`	Eigenes Programm starten
`Strg+B Strg+D`	Tmux in den Hintergrund schicken
`tmux a`	Tmux wieder öffnen und fortsetzen

- Mehr Infos zum in Ruhe ausprobieren unter `https://techfak.net/compute`

SSH Konfiguration

Konfigurationsdateien

Konfigurationsdateien in `~/.ssh`:

- authorized_keys
- config — Userspezifische Konfiguration
- id_ed25519
- id_ed25519.pub
- known_hosts — Hostkeys, denen man vertraut

SSH Konfiguration

Konfigurationsdateien

~/.ssh/config:

```
Host tfcompute
Hostname compute.techfak.de
User juser

Host myserver
Hostname asdf123.hetzner.de
User pinguin3000
IdentityFile id_ed25519_hetzner
Port 1234
```

ssh tfcompute ⇒ ssh juser@compute.techfak.de

ssh myserver ⇒ ssh pinguin3000@asdf123.hetzner.de -p 1234 -i /.ssh/id_ed25519_hetzner

SSH Login ohne Passwort

SSH Keypair

- Login mit Schlüsselpaar, ohne Passwort
- Besteht aus public key (endet auf `.pub`) und private key
- Private key ersetzt sozusagen das Passwort ⇒ geheim!
- Public key ist das Gegenstück (beim Passwort der Hash)
- Private key 'Schlüssel', public key 'Schloss'

Konfigurationsdateien in `~/.ssh`:

- authorized_keys public keys von zugehörigen private keys, die sich in diesen Account einloggen dürfen
- config
- id_ed25519 private key
- id_ed25519.pub public key
- known_hosts

SSH Login ohne Passwort

Erzeugen der SSH keys

Auf dem Unix System, von dem du dich später einloggen möchtest:

```
$ ssh-keygen -t ed25519
```

SSH Login ohne Passwort

Erzeugen der SSH keys

Auf dem Unix System, von dem du dich später einloggen möchtest:

```
 $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/juser/.ssh/id_ed25519):
```

SSH Login ohne Passwort

Erzeugen der SSH keys

Auf dem Unix System, von dem du dich später einloggen möchtest:

```
 $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/juser/.ssh/id_ed25519):

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

SSH Login ohne Passwort

Erzeugen der SSH keys

Auf dem Unix System, von dem du dich später einloggen möchtest:

```
 $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/juser/.ssh/id_ed25519):

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/juser/.ssh/id_ed25519.

Your public key has been saved in
/home/juser/.ssh/id_ed25519.pub .

The key fingerprint is:
d0:e7:f0:3c:78:5c:34:d6:57:c8:d5:26:18:b8:49:f0 juser@example
```

SSH Login ohne Passwort

Login mit Key

Jetzt **public** key zur ~/.ssh/authorized_keys hinzufügen.

```
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI[...]zG21v0/OhDnVPe juser@foobar
```

Auf dem Zielsystem anhängen an ~/.ssh/authorized_keys

SSH Login ohne Passwort

Im Netboot

Innerhalb des Netboots (z.B. von GZI auf compute) ist Login mit Key weder möglich, noch nötig.

- Der Grund dafür ist Kerberos (Schützt Homes vor unbefugtem Zugriff)
- ssh ohne Passwort ist aber möglich: `ssh -K compute`

SSH mit Keybpair ist trotzdem sinnvoll und weit verbreitet!
Z.B. von zu Hause (gleich), oder auf nicht Netboot Rechner.

SSH mit GUI

Fenster weiterleiten

ssh kann vom entfernten Rechner:

- Kommandozeile weiterleiten (Standard)
- Fenster weiterleiten (auf Wunsch)

```
> ssh -X login@shell.techfak.de
> ssh -X compute
```

Voraussetzung: Betriebssystem zu Hause ist

- Linux
- Mac OS X ab 10.5

SSH-Clients für Windows

SSH/SCP-Clients unter Windows

WinSCP: nur Dateien übertragen

`http://winscp.net/de`

Bitte **niemals** Paßwörter in WinSCP etc. speichern

- dort werden sie als erstes von Schadsoftware abgegriffen!
- auch wenn der Rechner erst Monate später infiziert wird

PuTTY: Dateien übertragen, Kommandozeile weiterleiten

`http://www.chiark.greenend.org.uk/~sgtatham/putty/`

Remote arbeiten

Jetzt von zu Hause

Compute ist praktisch, aber ohne weiteres nur aus der TechFak erreichbar...

- wie komme ich von zu Hause auf die TechFak-Rechner?
  - shell.techfak.de
- Datenaustausch von zu Hause mit dem TechFak-System
  - files.techfak.de

Remote arbeiten

Zwei Hosts: shell und files

`shell.techfak.de`

- Zum Arbeiten von zu Hause
- Temporäre Homes ⇒ Weiterverbinden auf compute
- Login mit Key (kein Passwort)

`files.techfak.de`

- Zum Datenaustausch
- „Remote-Home“
- Kein interaktiver Login möglich

⇒ Mehr Sicherheit

Remote arbeiten via shell

Von zu Hause am Uni-Rechner arbeiten

```
$ ssh ACCOUNT@shell.techfak.de
```

Login auf `shell` nur mit ssh-key ($\rightarrow$ kein Passwort).

Warum?

- Passwort raten („brute force“) nicht möglich
- Selbst mit kompromittiertem Passwort kein Login

Remote arbeiten via shell

Erzeugen der ssh-keys

Folgende Schritte sind notwendig:

1. ssh keypair erzeugen (auf [jedem] eigenen Rechner)
2. public key in die TechFak kopieren

Doku ausfürlich im Web: `https://techfak.net/remote/shell`

Remote arbeiten via shell

Upload des public keys

Erzeugen des Keys wie eben mit
`$ ssh-keygen -t ed25519`

Jetzt **public** key zur `~/.ssh/authorized_keys` hinzufügen.

ABER: Auf `shell` ist das Home nicht verfügbar!

⇒ `https://techfak.net/remote/shell/setup`

[Live Demo]

Remote arbeiten via shell

Upload des public keys

Erzeugen des Keys wie eben mit
`$ ssh-keygen -t ed25519`

Jetzt **public** key zur `~/.ssh/authorized_keys` hinzufügen.

ABER: Auf `shell` ist das Home nicht verfügbar!

⇒ `https://techfak.net/remote/shell/setup`

[Live Demo]

Geschafft!

Remote arbeiten via shell

Von zu Hause am Uni-Rechner arbeiten

Jetzt einloggen:

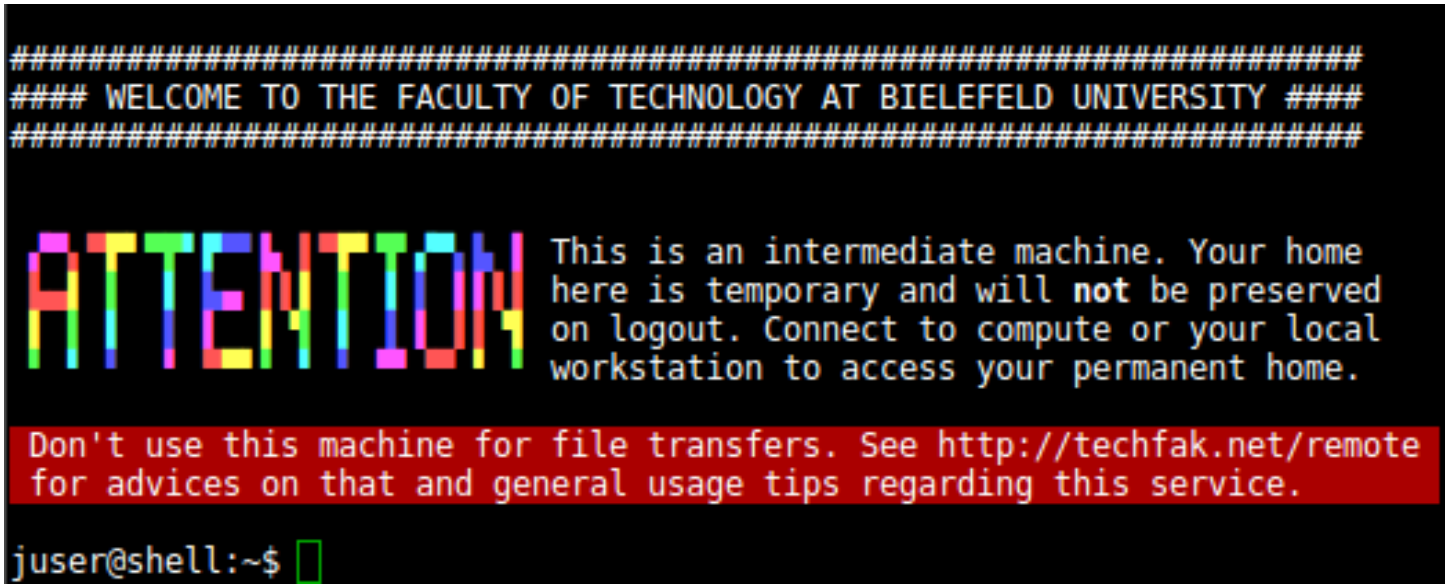
```
juser@foobar:~$ ssh juser@shell.techfak.de
The authenticity of host 'shell.techfak.de (2001:638:504:2041::226)'
can't be established.
ED25519 key fingerprint is
SHA256:0tsVGENxjW1Twqrg7FPQ6xrZ+e6ZcQ3rLU79+3I06Jo.
Are you sure you want to continue connecting (yes/no)?
```

Fingerabdruck und weitere Infos:

`https://techfak.net/remote/shell`

Remote arbeiten via shell

Geschafft!



Remote arbeiten via shell

Compute-Rechner

Shell hat:

- nur temporäre Homes.
- wenig Rechenleistung.

$\Rightarrow$ von dort per ssh auf die compute-Rechner weiterverbinden!

`juser@shell:~$ ssh compute` (Dieses mal mit GZI-Passwort!)

Daten übertragen via files

Was ist files?

Zweiter Host `files.techfak.de`:

- Login mit GZI Passwort
- Keine shell! (dafür ist `shell` da ..)
- Das Home ist euer remote Home! (nur von extern)

Daten übertragen via files

Was ist files?

Zweiter Host `files.techfak.de`:

- Login mit GZI Passwort
- Keine shell! (dafür ist `shell` da ..)
- Das Home ist euer remote Home! (nur von extern)

Remote Home:

- Liegt unter /media/remote/*juser*/
- Auch als Link im 'lokalen' Home: `remote`

Warum?

- Eure Daten sind sicherer
- Es ist schwerer den Account selbst mit Wissen um das Passwort(!) von ausserhalb zu übernehmen

Daten übertragen via files

Keys überprüfen

Immer wichtig!

```
juser@foobar:~$ scp testdatei juser@files.techfak.de:~/test
The authenticity of host 'files.techfak.de (2001:638:504:2041::230)'
can't be established.
ED25519 key fingerprint is
SHA256:qABq3ZWyOEjgVEnUdkIGE05Sfuy2dbSCh9FCsMM65Zc.
Are you sure you want to continue connecting (yes/no)?
```

Fingerabdruck und weitere Infos:

`https://techfak.net/remote/files`

Daten übertragen

von zu Hause auf Uni-Rechner

scp (secure copy)

```
$ scp datei juser@files.techfak.net:~/Ziel
```

Beispiele für Zielverzeichnisse:

:	Home-Verzeichnis (Remote!)
:~/ablage	Verzeichnis ablage im Remote-Home-Verzeichnis
:/tmp	Öffentliches lokales Verzeichnis auf dem Rechner

Daten übertragen

vom Uni-Rechner nach zu Hause

scp (secure copy)

```
$ scp juser@files.techfak.net:~/datei ~/Ziel
```

Wildcards sind möglich:

```
scp *.txt user@files.tech..
scp user@files.tech..:*.txt .
```

Für mehrere Dateien und/oder Verzeichnisse praktisch:

- `tar`-Archiv erzeugen und übertragen (`man tar`)

Alternativ könnt ihr `sftp` probieren.

Ende

Vielen Dank fürs Zuhören!

- Die Folien sind zum Nachlesen online
- `https://techfak.net` ist immer eine gute Idee!
- In der Fachschaft und beim Support (M3-107/100) findet ihr oft Menschen, die euch beim Linux installieren helfen :)