# SSH - Secure Shell

## TechTalk

Theresa Enghardt

http://wiki.freitagsrunde.org/Techtalks

13. April 2012

# SSH - Was ist das?

- **S**ecure **SH**ell
- Protokoll und Programm zum Fernzugriff auf Rechner
- Über die Kommandozeile (Shell)
- Muss SSH-Server laufen haben
- Befehle ausführen, Dateien anzeigen und ändern...
- ... und das alles verschlüsselt! (Sicher)

[1]

[1] von openssh.org

# Wozu brauche ich das?

- Datei liegt auf Uniserver und ihr kommt von zu Hause nicht ran
- Programm ist nur auf Uniserver installiert
- Rechner steht in nem Serverraum, da ist es laut und ungemütlich
- Zugriff auf weitere Rechner, zB dort im lokalen Netz
- Projekte im Studium
- Chatten, obwohl Ports geblockt sind (SOCKS-Proxy)

# Und wie mache ich das?

- Linux/Mac OS: OpenSSH client `ssh`
- Windows: PuTTY[2]
- zunächst mit Username und Passwort
- ... Sicher, dass es der **richtige** Username ist?

```
theresa@anghammarad:~$ ssh theri@furor.cs.tu-berlin.de
```

Erster SSH-Login

[2] `http://www.putty.org`

# Halt! Wer ist da? (Server)

```
theresa@anghammarad:~$ ssh theri@furor.cs.tu-berlin.de
The authenticity of host 'furor.cs.tu-berlin.de
    (130.149.17.59)' can't be established.
RSA key fingerprint is c8:c4:a7:19:cc:f5:22:87:c2:cf:c2:5d:
    a5:bb:d5:d5.
Are you sure you want to continue connecting (yes/no)?
```

Erster SSH-Login - Fingerprint

- Der Server versucht, sich auszuweisen, dass er wirklich furor ist
- Kryptographischer Schlüssel, der einen 'Fingerabdruck' **fingerprint** hat
- Schlüssel ist sehr lang - Fingerprint ist eine 'komprimierte' Fassung

# Halt! Wer ist da? (Server)

```
theresa@anghammarad:~$ ssh theri@furor.cs.tu-berlin.de
The authenticity of host 'furor.cs.tu-berlin.de
    (130.149.17.59)' can't be established.
RSA key fingerprint is c8:c4:a7:19:cc:f5:22:87:c2:cf:c2:5d:
    a5:bb:d5:d5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'furor.cs.tu-berlin.de
    ,130.149.17.59' (RSA) to the list of known hosts.
```

Erster SSH-Login - Fingerprint

- Idealerweise: Fingerprint steht auf Website oder auf Papier - abgleichen!
- Leider in der Praxis oft nicht machbar (steht nirgendwo)
- Theoretisch Gefahr, dass sich ein anderer Rechner als furor ausgibt

# Halt! Wer ist da? (Client)

```
theresa@anghammarad:~$ ssh theri@furor.cs.tu-berlin.de
The authenticity of host 'furor.cs.tu-berlin.de
    (130.149.17.59)' can't be established.
RSA key fingerprint is c8:c4:a7:19:cc:f5:22:87:c2:cf:c2:5d:
    a5:bb:d5:d5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'furor.cs.tu-berlin.de
    ,130.149.17.59' (RSA) to the list of known hosts.
theri@furor.cs.tu-berlin.de's password:
Linux furor 2.6.32-40-generic #87-Ubuntu SMP Tue Mar 6
    00:56:56 UTC 2012 x86_64 GNU/Linux
Ubuntu 10.04.4 LTS
[...]
theri@furor:~$
```

Erster SSH-Login - Passwort

# Ich bin drauf - und jetzt?

- z.B. Opal-Interpreter

```
theri@furor:~$ oasys
```

Opal-Interpreter ausführen

- z.B. Verzeichnisinhalt anzeigen

```
theri@furor:~$ ls
```

Verzeichnisinhalt anzeigen

- Wieder raus? Fenster zumachen oder `exit`

# SSH-Config

- Immer an den richtigen Usernamen denken nervt!
- Immer die volle URL eingeben nervt!
- **Konfigurationsdatei** speichert das für dich
- Linux/Mac OS: `/.ssh/config`
- d.h.: in deinem Homeverzeichnis im Ordner '.ssh' (versteckt)

SSH-Konfigurationsdatei

```
1 Host furor
2   Hostname furor.cs.tu-berlin.de
3   User theri
```

```
theresa@anghammarad:~$ ssh furor
```

SSH nun einfacher!

# Dateien kopieren

- Linux/Mac OS: Einfach auf der Kommandozeile per `scp` (Secure Copy)

```
theresa@anghammarad:~$ scp daten.txt furor:/home/theri/
    uni/
```

Kopieren per SCP

- Alternative: Furors Dateisystem lokal einbinden (mounten) und dann 'normal' kopieren
- Linux/Mac OS: SSHFS (**SSH F**ile**S**ystem)
- Windows: WinSCP[3]

```
theresa@anghammarad:~$ sshfs furor: ~/mnt
```

SSHFS einhängen (Verzeichnis mnt muss existieren!)

[3] http://www.winscp.net

# SSH-Keys

- Alternative zu Username und Passwort als Authentifizierung
- Immer zwei gehören zusammen:
  - Privater Schlüssel z.B. **id_rsa** - Geheim! Nicht weitergeben!
  - Öffentlicher Schlüssel z.B. **id_rsa.pub** - Weitergabe ungefährlich
- Öffentlicher Schlüssel auf den Server kopieren
- Privater Schlüssel bleibt auf dem eigenen Rechner
- Privater Schlüssel sollte mit Passphrase geschützt werden
- Erstellen mit `ssh-keygen` oder in Puttygen

# SSH-Keys

```
theresa@anghammarad:~$ ssh-keygen -b 2048 -f neuer_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in neuer_key.
Your public key has been saved in neuer_key.pub.
The key fingerprint is:
b2:c3:8f:ea:83:0b:ae:00:06:1f:af:b5:f7:ac:f0:11
    theresa@anghammarad
The key's randomart image is: [...]
```

SSH-Keys generieren

- Länge des Schlüssels: 2048 Bit reichen aus
- Benutzt eine Passphrase!

# SSH-Keys

```
theresa@anghammarad:~$ scp ~/.ssh/neuer_key.pub furor
theri@furor.cs.tu-berlin.de's password:
neuer_key.pub                 100%  401     0.4KB/s   00:00
theresa@anghammarad:~$ ssh furor
theri@furor:~$ mkdir .ssh
theri@furor:~$ cat neuer_key.pub >> .ssh/authorized_keys
```

SSH-Keys generieren

- Kopiert den **öffentlichen Schlüssel**, nicht den privaten!
- Geht auf den Server
- Hängt den Key an die Datei `authorized_keys` an

# SSH-Keys

```
theresa@anghammarad:~$ ssh furor -i ~/.ssh/neuer_key
```

SSH-Keys benutzen

- Automatisch den Key benutzen: Konfigurationsdatei ändern

SSH-Konfigurationsdatei

```
1 Host furor
2   Hostname furor.cs.tu-berlin.de
3   User theri
4   IdentityFile ~/.ssh/neuer_key
```

# SSH-Agent

- Dauernd Passwort oder Passphrase eingeben nervt!
- **ssh-agent** speichert die Passphrase eurer Keys
- Läuft unter aktuellen Linuxdistributionen meistens automatisch schon

```
theresa@anghammarad:~$ ssh-add ~/.ssh/neuer_key
Enter passphrase for /home/theresa/.ssh/neuer_key:
Identity added: /home/theresa/.ssh/neuer_key (/home/theresa
    /.ssh/neuer_key)
```

SSH-Agent benutzen

- Beim nächsten SSH (oder SCP, oder SSHFS...) müsst ihr keine Passphrase mehr eingeben

# SOCKS-Proxy

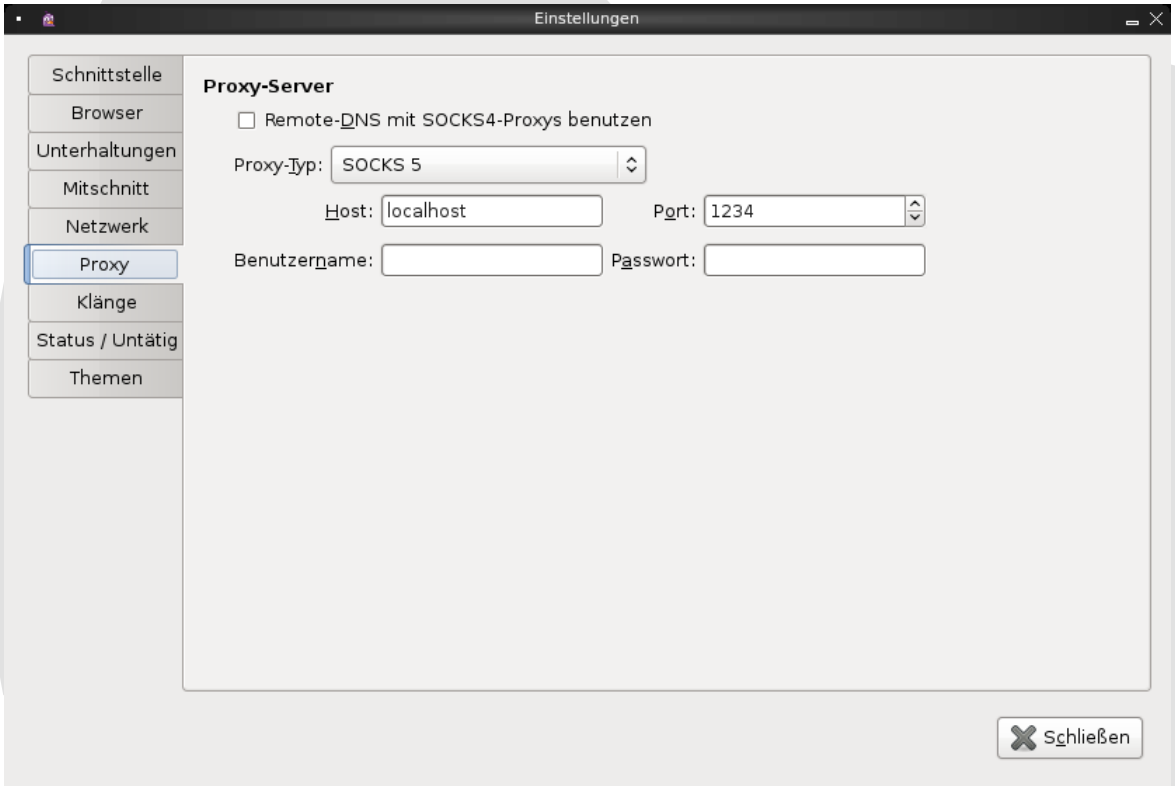
- Über die SSH-Verbindung was anderes tunneln, z.B. IRC
- Ist zumindest bis zum SSH-Server verschlüsselt
- Falls IRC geblockt wird, aber SSH nicht, kommt es durch
- Lokaler Port $>$ 1024 wird über SSH-Server getunnelt

```
theresa@anghammarad:~$ ssh -D 1234 furor
```

SOCKS-Proxy

# SOCKS-Proxy

- In Pidgin, IRC-Client u.ä. einen SOCKS-Proxy einstellen
- Host: `localhost`, Port: `1234` (bzw den, den ihr eingegeben habt)
- kein Username, Passwort o.ä.: Proxy läuft auf eurem lokalen Rechner

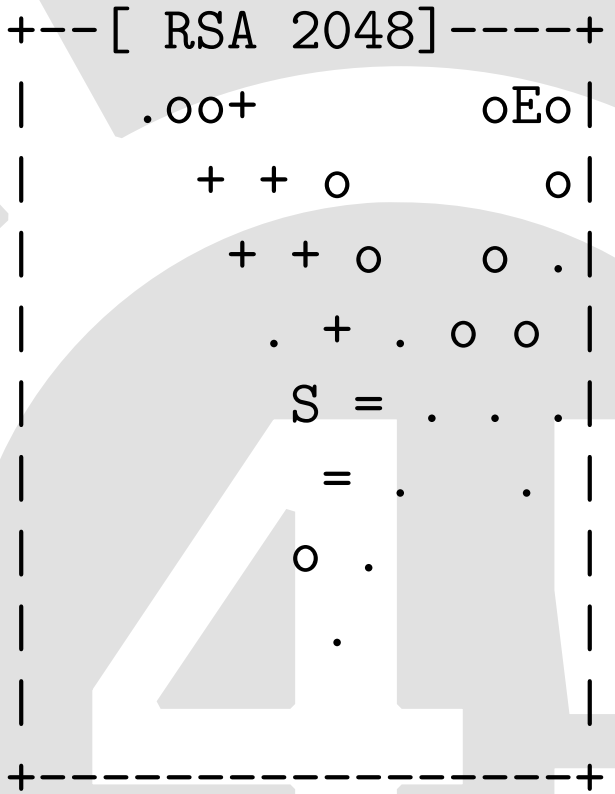


# Weitere tolle Tricks

- VisualHostKey statt des Fingerprints
- Leichter zu verifizieren
- In der SSH-Config eintragen:

VisualHostKey

```
1 VisualHostKey true
```



# Weitere tolle Tricks

- SSH auf einen Host, der als ‘Gateway’ fungiert
- nur von diesem Gateway aus kommt ihr auf die anderen Hosts
- Lösung: Host in die SSH-Config eintragen und ProxyCommand eintragen
- Kommando: ‘ssh dich auf gateway und leite alle Ports weiter’
- auf beiden Hosts kann/sollte ein SSH-Key liegen

ProxyCommand

```
1 ProxyCommand ssh gateway -W %h:%p
```

# Und ihr so?

Folien:
`http://docs.freitagsrunde.org/Veranstaltungen/techtalk/2012/ssh_slides.pdf`

Theresa Enghardt
`theresa@freitagsrunde.org`