

# Netzwerk

Hans Buchmann FHNW/IME

13. Oktober 2014

# Ziel

## RaspberryPi am Schulnetz

- ▶ verschiedenen Netze

	public	vpn	eduroam
LAN	x	x	
WLAN	x	x	x

- ▶ Sicherheit
  - ▶ keine sichtbaren Passworte

Wichtig  
Wichtig

- ▶ **richtiges** Passwort auf **RaspberryPi**
- ▶ `passwd`

# Probleme

## Konfiguration mit restriktivem Netzzugriff

	public	vpn	eduroam
LAN	<i>local</i>	<i>full</i>	
WLAN	<i>local</i>	<i>full</i>	<i>full</i>

**local** Zugriff nur im lokalen Netz

**full** voller Zugriff nach Konfiguration

Was haben wir ?

Verbindung zum **RaspberryPi**

*Host offline* mount SD-Card

RS232 online wichtige Verbindung

# Die Schritte Bootstrapping

- ▶ Ethernet Verbindung *Host* - **RaspberryPi**
- ▶ **RaspberryPi** Installation der notwendigen *packages*
  - ▶ vom Internet mit *Host* als gateway

## ssh & Co

- ▶ die Stärken von UNIX ssh nutzen
  - ▶ `glbs/8-ssh/doc/ssh.pdf`

## Die Schritte

- ▶ LAN **RaspberryPi** *Host*

  - `ssh` für die Verbindung

  - `sshfs` für die einfache Konfiguration Files editieren

- ▶ WLAN public
- ▶ WLAN vpn
- ▶ WLAN eduroam



# LAN

## Host RaspberryPi

*Host* eth0 Schnittstelle

▶ `ifconfig eth0 192.168.1.1 up`

**RaspberryPi** eth0 Schnittstelle

▶ `ifconfig eth0 192.168.1.2 up`

*test* mit `ping`

## ssh/sshfs RaspberryPi

```
ssh user@ip
```

▶ ohne Passwort → ssh-copy-id

```
sshfs sshfs user@ip:dir dir
```

# pacman

Der *packetmanager*

## ► Die Files

- /etc/pacman.conf

```
...  
#XferCommand = /usr/bin/curl -C - -f %u > %o  
#XferCommand = /usr/bin/wget --passive-ftp -c -O %o %u  
..
```

- /etc/pacman.d/mirrorlist

## wget für das Internet

*Host* Probieren

- ▶ wget

**RaspberryPi** Mit ssh Tunnel

- ▶ ssh user@host 'wget ...' > file  
    > redirect

## RaspberryPi `ssh` Tunnel

Alles ist ein File

- ▶ `ssh Host cmd cmd` auf dem host ausführen
- ▶ `stdout` von `cmd` auf **RaspberryPi**

# Bootstrap

*Host* sshfs **RaspberryPi** *Host*

*Host* **RaspberryPi** /etc/pacman.conf editieren

**RaspberryPi** pacman -Suy

```
#XferCommand = /usr/bin/curl -C - -f %u > %o
```

```
#XferCommand = /usr/bin/wget --no-check-certificate --passive-ftp
```

```
XferCommand = ssh user@ip 'wget -O - -c %u' > %o
```

## ip forwarding provisorisch

- ▶ [https://wiki.archlinux.org/index.php/Internet\\_sharing](https://wiki.archlinux.org/index.php/Internet_sharing)
- ▶ Testen ob das *forwarding* wirklich funktioniert

# Die Probleme

- ▶ die Administration ist nicht ganz einfach
- ▶ es gibt verschiedene *tools*:
  - ▶ netctl
  - ▶ wicd
  - ▶ NetworkManager
  - ▶ ...
- ▶ verschiedene Verfahren für eine sichere Verbindung
  - ▶ IEEE 802.11i

## Remark(s):

- ▶ es sollte nur ein *tool* laufen
- ▶ die *tools* verwalten LAN und WLAN



# geschichtete Software

## vereinfachtes Bild

administration	netctl & Co.
WLAN Konfiguration	wpa_supplicant
Hardware	

## wpa\_supplicant der WLAN der Bittsteller

- ▶ zwei Komponenten

Prozess `wpa_supplicant` *options*

- ▶ Konfiguration typ.

Bedienung `wpa_cli` <sup>1</sup>

---

<sup>1</sup>`cli` command line interface

# VPN

## Konfiguration

```
IPSec gateway vpn.fhnw.ch  
IPSec secret 29HeiowA  
IPSec ID STU-03-001  
IKE Authmode psk  
Xauth username kkkkk  
Xauth obfuscated password iii
```

# Die Sache mit obfuscated Vernebeln

- ▶ das Programm `obfuscate.c`
- ▶ das Programm `cisco-decrypt`

ssh  
ssh mount

- ▶ ssh ohne Passwort `ssh-copy-id`
  - ▶ Kopiere **public key** vom *Host* auf **RaspberryPi**
- ▶ `sshfs`
  - ▶ *target-root* auf *Host* sichtbar
  - ▶ `fusermount -u` für *unmount*

ssh

## Vorbereitungen RaspberryPi

- ▶ Setze Datum/Zeit date
  - ▶ Gewisse Zertifikate schauen auf das Datum
- ▶ Konfiguriere /etc/pacman.conf für ssh Tunnel:  

```
XferCommand = ssh user@ip 'wget -O - -c %u' > %o
```

  - ▶ via *Host*
- ▶ update
  - ▶ `pacman -Suy`
- ▶ `vpnc` package
  - ▶ `package -Su vpnc`

## Prozess/CLI

- ▶ Prozess `wpa_supplicant` *options*
- ▶ CLI `wpa_cli`
  - ▶ wie funktioniert das `cli`
- ▶ WLAN `fhnw-public`
  - ▶ `dhcpcd` für eine `ip` Adresse
- ▶ Konfigurationsfile
- ▶ VPN
  - ▶ mit `vpnc`
- ▶ eduroam
  - ▶ mit `swpa_supplicant`

## Nützliche Files und Adressen

host-success.txt

wpa\*

<https://cat-test.eduroam.org>

log für eduroam

Konfiguration

sollte gehen <sup>2</sup>

---

<sup>2</sup><https://cat.eduroam.org> funktioniert nicht



## Zwei Verbindungen:LAN/WLAN für RaspberryPi

WLAN mit der Welt

- ▶ **RaspberryPi** ist Client

LAN mit den *Host* im gleichen Netz

- ▶ **RaspberryPi** ist Server

# LAN

## *Host*

- ▶ `ifconfig eth0:0 192.168.1.1 up`

Remark: die anderen Verbindungen bleiben

## RaspberryPi

- ▶ `ifconfig eth0 192.168.1.2 up`

# WLAN RaspberryPi

## Manuell

Background `wpa_supplicant -D wext -i wlan0 -cwpa.conf -B`

Foreground `wpa_cli`

# WLAN RaspberryPi

netctl

- ▶ <https://wiki.archlinux.org/index.php/netctl>
- ▶ Passwort mindestens verschleiern