



## SV3: Linux Server

Fernwartung

# Agenda

## Auf anderen Linux-Rechnern arbeiten

- SSH
- SCP /SFTP
- Curl
- Vnc

# SSH – secure shell

Um mit anderen Rechnern zu kommunizieren, benötigen wir neben Hardware und den IP-Nummern auch entsprechende Softwarepakete.

- **ssh (secure shell)**, die sichere Methode, um eine Shell zur Ausführung anderer Programme auf einem entfernten Rechner im Netz zu starten bzw. sich an einem entfernten Rechner anzumelden.
- **rsh (remote shell)**, **rlogin (remote login)** oder **telnet (telecommunication network)** sind ältere ungesicherte Kommandos und sollten nicht mehr verwendet werden, denn sie übertragen auch das Passwort so, dass es von entsprechender »**Lauschsoftware**« abgefangen werden kann.

# SSH – secure shell

**Server, die nicht physikalisch vor Ihnen stehen, können Sie nur über eine Netzwerkverbindung administrieren.**

- Das bevorzugte Werkzeug hierfür ist **SSH (Secure Shell)**. Es ermöglicht sowohl die einfache Ausführung von Kommandos als auch die Bedienung grafischer Programme über eine sichere Netzwerkverbindung.
- Die einzige Voraussetzung besteht darin, dass auf dem Server ein **SSH-Server** installiert ist.

## Arbeiten mit ssh

Um eine Verbindung über **ssh**\* mit einem entfernten Rechner aufzunehmen, müssen wir dort einen gültigen Benutzernamen und das Passwort dazu kennen.

- Außerdem wird das erste Mal ein sog. »**Fingerprint**« zwischen den beiden Rechnern ausgetauscht, der bestätigt werden muss.

```
ssh [Optionen] [Name@]Rechnername
```

# SSH – secure shell

## ssh – Kommando, um netzwerkweit verschlüsselt zu arbeiten

```
sb@ub:~$ ssh sb@server
The authenticity of host 'server (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:uqSwQQ67+6FFv3/k2M40jDLDkIB5rQh3+HoZiz7afRM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server' (ECDSA) to the list of known hosts.
sb@server's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 5.3.0-46-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
```

## Konfiguration und Absicherung

Die Konfigurationsdateien zu **sshd** befinden sich im Verzeichnis **/etc/ssh**. Für die Server-Konfiguration ist **sshd\_config** zuständig.

- Normalerweise kann diese Datei unverändert bleiben, d. h., der **SSH-Server** sollte auf Anhieb funktionieren.
- Die Kommunikation erfolgt standardmäßig über den **IP-Port 22**.

## Konfiguration und Absicherung

Grundsätzlich läuft der SSH-Server auf Anhieb ohne Konfigurationsarbeit. Das ist Absicherung allerdings ein nicht zu unterschätzendes Sicherheitsrisiko:

- Cracker verwenden automatisierte Tools, die im Internet nach Servern suchen und sich dort einzuloggen versuchen.
- (z.b häufig betroffen der Raspberry Pi da Standard User & Passwort nicht geändert werden und der Port 22 im Router einfach freigeben wird.)



## Konfiguration und Absicherung

Die folgenden Maßnahmen reduzieren jeweils die Wahrscheinlichkeit eines Crack-Angriffs auf Ihren SSH-Server.

- Sie können einzeln oder in Kombination angewendet werden.

# SSH – secure shell


## Konfiguration und Absicherung

Ein Angreifer möchte root-Rechte erzielen – und am einfachsten gelingt das natürlich durch einen root-Login.

- Dabei muss nur ein Parameter (das root-Passwort) erraten werden. Wesentlich sicherer ist es, einen direkten root-Login via SSH zu verbieten.

```
# Authentication:
```

```
LoginGraceTime 2m  
PermitRootLogin no  
StrictModes yes  
MaxAuthTries 6  
MaxSessions 10
```



## Konfiguration und Absicherung

Damit die Änderung wirksam wird, müssen Sie sshd dazu auffordern, die Konfigurationsdateien neu einzulesen:

```
# systemctl reload sshd
```

- Für den Angreifer hat das die Konsequenz, dass nun zwei Parameter unbekannt sind:
- der Login-Name und das Passwort

# SSH – secure shell

## Konfiguration und Absicherung

Der SSH-Server kommuniziert standardmäßig über den **Port 22**. Mit der Port-Zeile können Sie mühelos einen anderen, momentan unbenutzten Port einstellen.

```
# The strategy used for options in the default sshd_config shipped with  
# OpenSSH is to specify options with their default value where  
# possible, but leave them commented. Uncommented options override the  
# default value.  
  
Port 6600
```

Da viele automatisierte Crack-Tools nur den **Port 22** berücksichtigen, vermeiden Sie auf einen Schlag viele Sicherheitsprobleme.

## Fail2Ban

ist gewissermaßen ein moderner Nachfolger von **DenyHosts** und unterscheidet sich in zwei Punkten von **DenyHosts**:

- Zum einen kann das Programm neben SSH auch diverse andere Netzwerkdienste überwachen und bei Bedarf für einzelne Hosts blockieren.
- Zum anderen erfolgt die Blockade nicht durch Einträge in `/etc/denyhosts.conf`, sondern durch Firewall-Regeln.

## Fail2Ban

Nach der Installation des fail2ban-Pakets erstellen Sie im Verzeichnis `/etc/fail2ban` eine Kopie der vorgegebenen Konfigurationsdatei `jail.conf` unter dem Namen `jail.local`.

- Dort führen Sie alle weiteren Änderungen durch. Diese Änderungen haben nun Vorrang gegenüber den Grundeinstellungen in `jail.conf`.

# SSH – secure shell

## Als »Jail« bezeichnet Fail2Ban einen zu überwachenden Dienst.

Die mitgelieferten Konfigurationsdateien in `/etc/fail2ban` enthalten Regeln für alle möglichen Dienste (Jails); standardmäßig sind diese aber alle deaktiviert.

- Um die SSH-Regeln zu bearbeiten, suchen Sie in `jail.local` nach dem Abschnitt `[sshd]`:

```
[sshd]
port    = ssh # Falls der Port geändert wurde, muss dies auch hier erfolgen!
logpath = %(sshd_log)s
backend = %(sshd_backend)s
maxretry = 5
bantime  = 10m
```

Die Default Konfiguration von Fail2Ban sieht vor, dass ein Host nach fünf vergeblichen SSH-Login-Versuchen innerhalb von zehn Minuten für zehn Minuten gesperrt wird.

## **Sicherheit bei SSH, weitere Möglichkeiten sind:**

### **Authentifizierung mit Schlüsseln**

Am sichersten ist die Verwendung des SSH-Servers, wenn Sie sich nicht mit einem Schlüssel erzeugen Passwort authentifizieren, sondern mit einem Schlüssel.



## Sicherheit bei SSH, weitere Möglichkeiten sind:

### Login mit dem Google-Authenticator

Eine weitere Variante besteht darin, eine Zwei-Faktor-Authentifizierung einzurichten.

- Bei jedem Login müssen Sie zusätzlich zum regulären Passwort einen nur für kurze Zeit gültigen Einmal-Code eingeben. Dieser Code wird von einer Smartphone-App generiert.

## Kopieren von Dateien im Netz

Auch bei scp müssen die korrespondierenden Rechner vorab sog. Fingerprints austauschen, die bestätigt werden müssen. Ist der Austausch bereits durch ssh erfolgt, ist kein weiterer Austausch mehr erforderlich.

**scp Quelle[Rechnername:]/Pfad/Datei Ziel[Rechnername:]/Pfad/Datei oder Verzeichnis**

```
sb@ub:~$ scp geheimatedaten.txt sb@server:/home/sb
sb@server's password:
geheimatedaten.txt          100%   0   0.0KB/s   00:00
sb@ub:~$
```

# curl – command line tool and library

## cURL

ist ein Programm, das es ermöglicht, ohne Benutzerinteraktion Dateien von oder zu einem Server zu übertragen.

- Neben HTTP unterstützt das Programm noch eine Vielzahl von weiteren Netzwerkprotokollen wie:
  - FTP, FTPS, HTTPS, GOPHER, TELNET, DICT, FILE und LDAP.
- Die Steuerung erfolgt über Kommandozeilenparameter, die beim Programmaufruf angegeben werden.

# curl – command line tool and library

## Anwendungen - 1. Websites auslesen

Mit diesem Kommando wird die Datei index.html ausgelesen und auf der Standardausgabe ausgegeben.

```
curl http://example.com/index.html
```

Möchten wir die Datei unter dem Namen savedpage.html abspeichern, benutzen wir dazu folgenden Befehl:

```
curl -o savedpage.html http://example.com/index.html
```

Erfordert eine Website die Anmeldung über den HTTP-Authentifizierungsmechanismus, lassen sich über cURL auch Benutzername und Passwort übergeben.

```
curl -u username:password http://example.com/index.html
```

# curl – command line tool and library

## Anwendungen - 2. URL mit Variablen GET-Parameter auslesen

cURL ermöglicht die Übergabe eines regulären Ausdrucks als GET-Parameter. Nehmen wir an eine Website wäre über ein CMS betrieben und böte die einzelnen Dokumente über URLs nach folgenden Schema an.

```
http://example.com/pages.php?id=1  
http://example.com/pages.php?id=14  
http://example.com/pages.php?=78
```

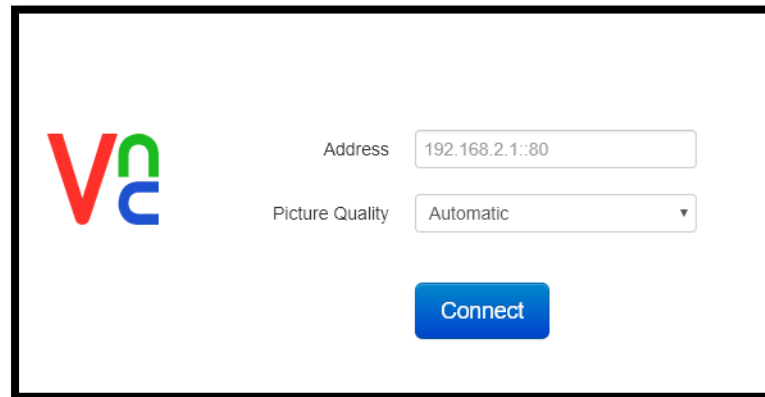
An cURL lässt sich der Parameter id als regulärer Ausdruck übergeben, um alle ids und damit Dokumente in einen bestimmten Bereich zu erfassen.

```
curl -o pages#1.html http://example.com/pages.php?id=[1-99]
```

# vnc - Virtual Network Computing

## VNC steht für Virtual Network Computing

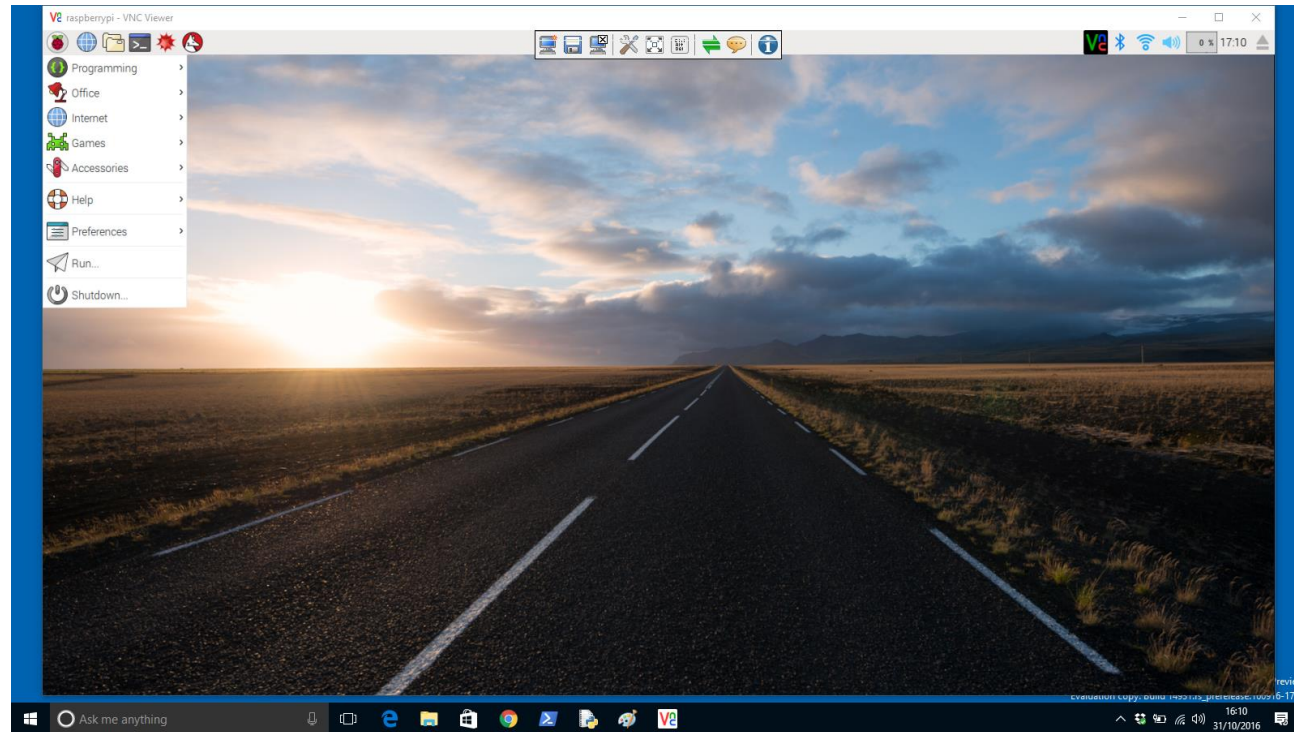
Ist eine Software, um den Bildschirminhalt über ein lokales Netzwerk zu übertragen und von einem anderen Rechner aus zu steuern.



VNC steht nicht nur unter X bzw. Linux, sondern auch für die meisten anderen Betriebssysteme zur Verfügung, inklusive Microsoft Windows.

# vnc - Virtual Network Computing

## VNC-Sitzung: Windows 10 zu Raspberry Pi



# vnc - Virtual Network Computing

## Bildschirmfreigabe

Solange das Programm läuft, kann nun mit einem VNC-Client der aktuelle Desktop unter der Adresse hostname:5900 ferngesteuert werden.

```
user$ x11vnc -usepw -forever -display :0  
Enter VNC password: *****  
Verify password:      *****  
Write password to .vnc/passwd? y
```

Beim Verbindungsaufbau ist dasselbe Passwort erforderlich, das Sie beim ersten Start von x11vnc zweimal angeben müssen.



## VNC-Server

Beim traditionellen Weg lauscht der VNC-Server auf TCP-Port 5900. Er bietet dort sein Display 0 an. Weitere Displays können über zusätzliche Ports angeboten werden.

- Diese sind meistens 590n, wobei n die Displaynummer ist. Falls der Host hinter einer Firewall (z.B. einem Hardware-Router) erreichbar sein soll, müsste der auf den entsprechende TCP-Port weitergeleitet werden.
- Bei von außen (= über das Internet) erreichbaren Rechnern wäre dies ohne verschlüsselte Verbindung grob fahrlässig. Für die einfache Anwenderunterstützung ist dieser Weg daher nicht zu empfehlen.

# vnc - Virtual Network Computing

## Hinweis:

- VNC ist ein altes und unsicheres Verfahren, denn die Übertragung der Daten erfolgt Verschlüsselung unverschlüsselt.
- Wenn Sie Wert auf mehr Sicherheit legen, müssen Sie den VNC Datenstrom über einen verschlüsselten Tunnel leiten oder VNC-Implementierungen mit integrierter Verschlüsselung nutzen.
- Unkompliziert ist die erste Variante:
  - Dabei konfigurieren Sie den VNC-Server so, dass er ausschließlich Verbindungen vom lokalen Rechner (localhost) zulässt.



**VIELEN DANK  
FÜR IHRE  
AUFMERKSAMKEIT!**