HOCHSCHULE MITTWEIDA
University of Applied Sciences

Bundeskriminalamt

Betriebssysteme

Linux Sicherheit

Autor: Felix Fischer, M.Sc.

Referent: Tim Wetterau, B.Sc.

Stand: 15.06.2023

hossfeld@hs-mittweida.de

Agenda

1. Grundprinzipien
2. Programmüberwachung
3. Netzwerksicherheit
4. Rechteverwaltung
5. Fernzugriff
6. Dateisicherheit

Grundprinzipien

der IT-Sicherheit

Sicherheit Grundprinzipien

Aus der Praxis

- Nur so viel, wie nötig
  - Angriffsfläche reduzieren
- Verteidigung in der Tiefe
  - Redundanzen der Sicherheit
- Kenne deine Systeme und Infrastruktur
  - Was ist installiert?
  - Wer nutzt was?
  - Dokumentiere!
- Überprüfe deine Sicherheit
  - Unerwartetes?
  - Welche Angriffsszenarien?

Theoriebegriffe

- CIA
  - Vertraulichkeit (confidentiality)
    - Zugriff nur von Berechtigten
  - Integrität (integrity)
    - Schutz vor unbemerkter Veränderung
  - Verfügbarkeit (availability)
    - Ausfälle vermeiden
- AAA
  - Authentifizierung
    - Benutzeridentifikation
  - Autorisierung
    - Was darf dieser Benutzer?
  - Accounting
    - Was macht dieser Benutzer?

Kerckhoffs' Prinzip (1883)

Algorithmus-Geheimhaltung

- Security by Obscurity (Sicherheit durch Verschleierung)
- Neuer Algorithmus erfordert viel Arbeit
- Algorithmus durch Reverse Engineering ermittelbar

→ Schlechte Praxis

Sicherheit durch Geheimhaltung des Schlüssels

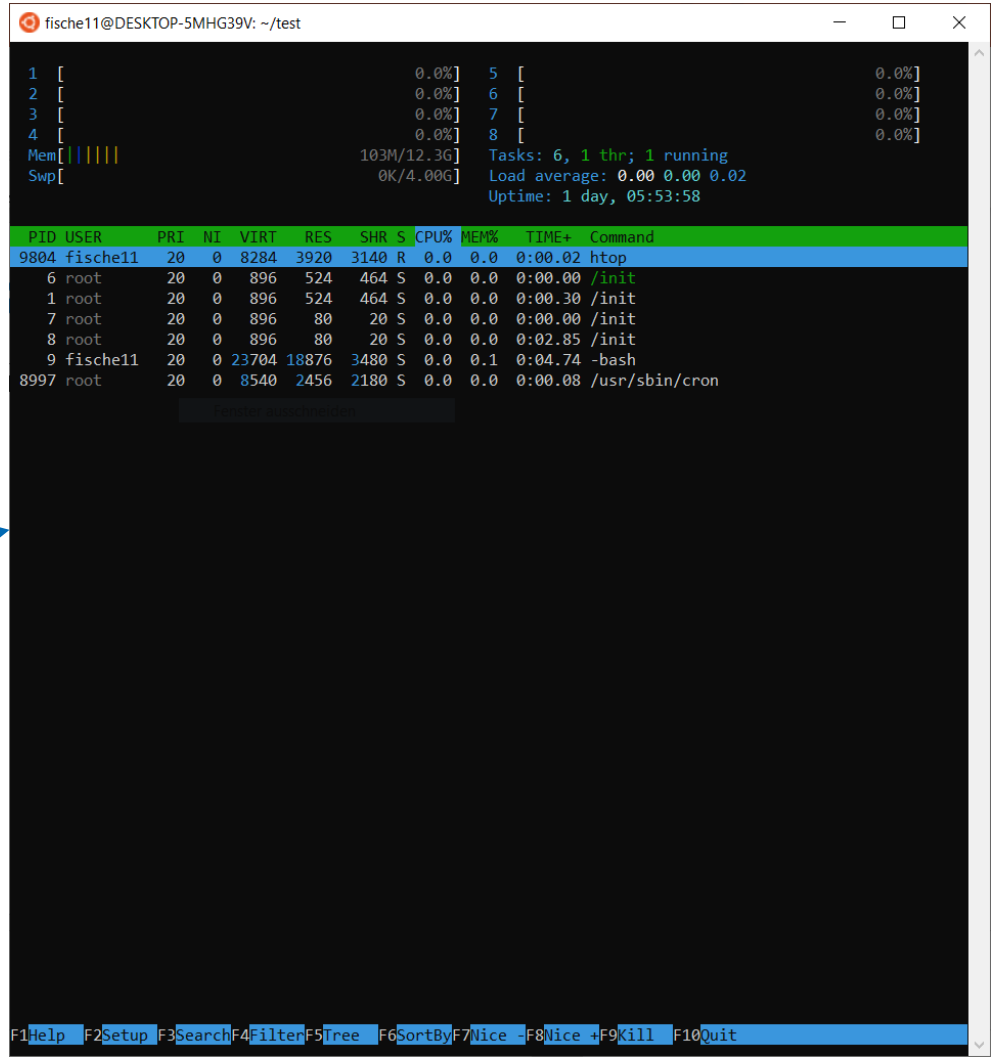
- Schlüsselgenerierung einfach
- schneller Austausch möglich
- Schlüssel tauschen ist einfacher als Algorithmus tauschen.
- Verschlüsselungsalgorithmus kann jedem bekannt sein

→ Grundsatz moderner Kryptographie

Programmüberwachung

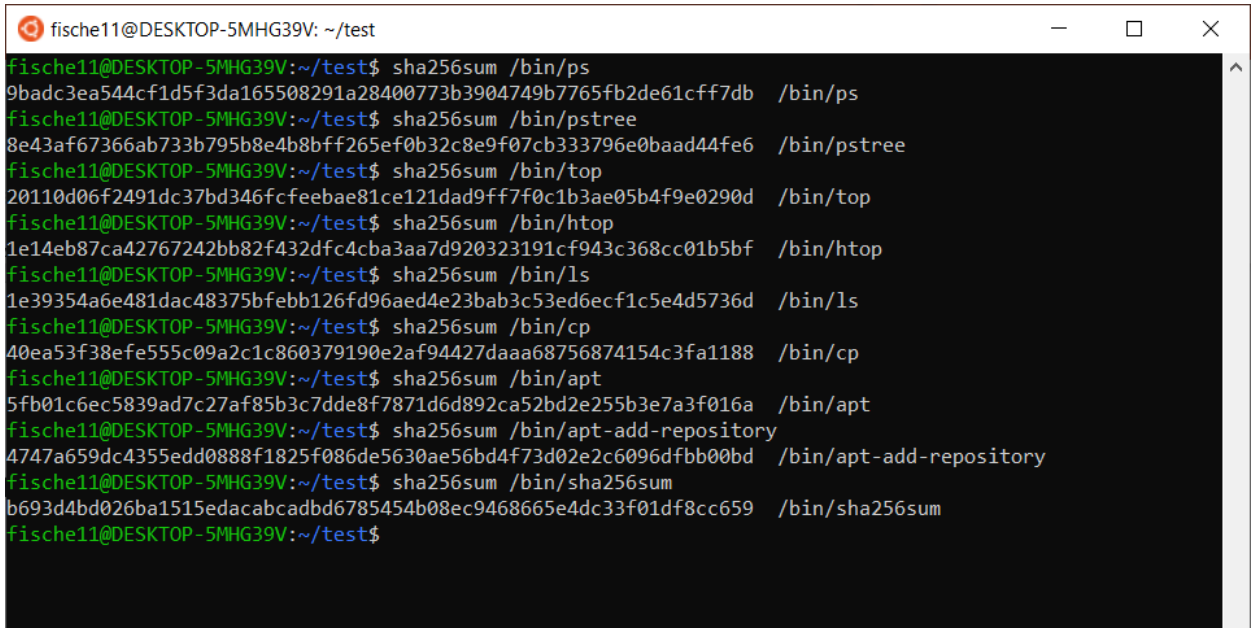
Laufende Programme anzeigen

- ps –aux
  - Alle Prozesse anzeigen
- pstree
  - Startabhängigkeit / -reihenfolge von Prozessen anzeigen
- top
  - „Taskmanager“
- htop
  - Verbesserte Version von top
- Systemüberwachung in GUI



Programmhash

- Angriffszenario:
  - ps, top, … wurde manipuliert
  - blendet Schadsoftware aus
- Veränderung von Programmen erkennen
- sha256sum <Programm>
- sha256sum /bin/*
- Regelmäßig Hash Prüfen (Skript über cron)
- **Achtung!**
  - Patch ändert auch Hashwert



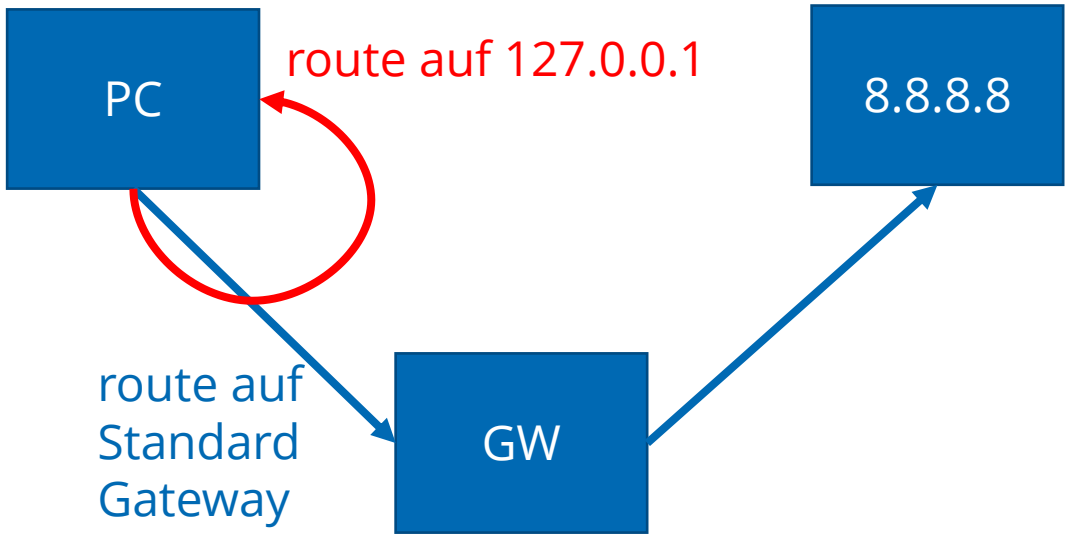
Updates

- Warum Updates?
  - Sicherheitsempfehlung Nr. 1
- Updates entfernen
  - Fehler
  - Sicherheitslücken
- Automatisierte Updates
  - Unattended-Upgrade
  - Cron-jobs
- Updates über Paketmanager
  - apt update
  - apt upgrade
  - apt dist-upgrade
  - apt autoremove
  - apt autoclean
- Manuell installierte Programme
  - Auf neue Version prüfen
  - Gegenfalls manuell aktualisieren

Netzwerksicherheit

IP-Filter mit route

- route als Ausgangsfilter
- Feste Routingtabelle als Filter
- Redundanz durch IP-Tables




```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         DESKTOP-5MHG39V 0.0.0.0         UG    0      0        0 eth0
172.29.64.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=19.4 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 19.404/19.404/19.404/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$ sudo route add -host 8.8.8.8 metric 10 dev lo
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         DESKTOP-5MHG39V 0.0.0.0         UG    0      0        0 eth0
dns.google      0.0.0.0         255.255.255.255 UH    10     0        0 lo
172.29.64.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

fische11@DESKTOP-5MHG39V:~/test$ sudo route del 8.8.8.8
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         DESKTOP-5MHG39V 0.0.0.0         UG    0      0        0 eth0
172.29.64.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=24.7 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.714/24.714/24.714/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$
```

DNS-Filter mit hosts

- DNS Auflösung
  - Definiert in /etc/nsswitch.conf
  - Standard:
    - DNS in /etc/hosts-Datei
    - DNS im Cache
    - DNS über Server abfragen
- Eintrag in Hosts:
  - 127.0.0.1 zu_sperrende_Domain
- **Achtung!**
  - Subdomain wird nicht gesperrt
  - Wildcards funktionieren nicht

```
linuxmint@linuxmint-VirtualBox:~$ tail -n 2 /etc/hosts
127.0.0.1 google.at

linuxmint@linuxmint-VirtualBox:~$ nslookup google.at
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.at
Address: 127.0.0.1

linuxmint@linuxmint-VirtualBox:~$ nslookup google.de
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.de
Address: 142.250.185.131
Name:   google.de
Address: 2a00:1450:4001:803::2003

linuxmint@linuxmint-VirtualBox:~$ ping -c 1 google.at
PING google.at (127.0.0.1) 56(84) Bytes Daten.
64 Bytes von localhost (127.0.0.1): icmp_seq=1 ttl=64 Zeit=0.032 ms

--- google.at ping-Statistik ---
1 Pakete übertragen, 1 empfangen, 0% Paketverlust, Zeit 0ms
rtt min/avg/max/mdev = 0.032/0.032/0.032/0.000 ms
linuxmint@linuxmint-VirtualBox:~$ ping -c 1 google.de
PING google.de (142.250.185.131) 56(84) Bytes Daten.
64 Bytes von fra16s50-in-f3.1e100.net (142.250.185.131): icmp_seq=1 ttl=56 Zeit=21.3 ms

--- google.de ping-Statistik ---
1 Pakete übertragen, 1 empfangen, 0% Paketverlust, Zeit 0ms
rtt min/avg/max/mdev = 21.336/21.336/21.336/0.000 ms
linuxmint@linuxmint-VirtualBox:~$ nslookup www.google.at
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.at
Address: 216.58.212.163
Name:   www.google.at
Address: 2a00:1450:4001:802::2003
```

Offene Ports

- Offener Port = Öffnung im Haus
- Nicht benötigte Ports schließen
- Offene Ports absichern
  - Authentifizierung
  - Welches Programm hinter Port?
  - Worauf hat Programm Zugriff?
- netstat –tulpen
- nmap –A 127.0.0.1

```
linuxmint@linuxmint-VirtualBox:~$ sudo netstat -tulpen
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Benutzer   Inode      PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      0          23266      699/cupsd
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      101        18422      543/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      0          24715      739/sshd: /usr/sbin
tcp6       0      0 ::1:631                 :::*                    LISTEN      0          23265      699/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      0          24717      739/sshd: /usr/sbin
udp        0      0 127.0.0.53:53           0.0.0.0:*                           101        18421      543/systemd-resolve
udp        0      0 0.0.0.0:35100           0.0.0.0:*                           119        23879      564/avahi-daemon: r
udp        0      0 0.0.0.0:631             0.0.0.0:*                           0          24588      676/cups-browsed
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           119        23877      564/avahi-daemon: r
udp6       0      0 :::46186                :::*                                119        23880      564/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                119        23878      564/avahi-daemon: r
linuxmint@linuxmint-VirtualBox:~$ sudo nmap -A 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-18 12:11 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000079s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
631/tcp open  ipp     CUPS 2.3
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: CUPS/2.3 IPP/2.1
|_http-title: Home - CUPS 2.3.1
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.10 seconds
linuxmint@linuxmint-VirtualBox:~$ sudo nmap -A -6 ::1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-18 12:11 CEST
Nmap scan report for ip6-localhost (::1)
Host is up (0.0000080s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
631/tcp open  ipp     CUPS 2.3
|_http-server-header: CUPS/2.3 IPP/2.1
|_http-title: Ung\xC3\xBCltige Anfrage - CUPS v2.3.1
Device type: general purpose
Running: Linux 3.X|4.X
```

iptables

- Firewallregeln
- iptables -L -v
  - Regeln anzeigen mit Details
- INPUT
  - Regeln für eingehende Pakete
- FORWARD
  - Regeln für weitergeleitete Pakete
- OUTPUT
  - Regeln für ausgehende Pakete
- ACCEPT
  - Wird durchgelassen
- DROP
  - Wird verworfen
- REJECT
  - Wird verworfen und Sender informiert

iptables-Regel hinzufügen

- `sudo iptables`

`-A <Ankunftsliste>`	→ `INPUT, FORWARD, OUTPUT`
`-i <Schnittstelle>`	→ `Netzwerkschnittstellennamen`
`-p <Protokoll>`	→ `tcp, udp, icmp, …`
`-s <Quelle>`	→ `Quell-IP`
`--dport <Ziel-Port>`	→ `Portnummer von Ziel-Port`
`-j <Aktion>`	→ `ACCEPT, DROP, REJECT`

iptables-Regel hinzufügen

- sudo iptables
  -A INPUT
  -i eth0
  -p tcp
  --dport 80
  -j ACCEPT

- sudo iptables
  -A INPUT
  -i wlan0
  -p tcp
  --dport 22
  -j ACCEPT

- sudo iptables
  -A FORWARD
  -i tun0
  -s 192.168.0.1
  -j REJECT

iptables-Regel löschen

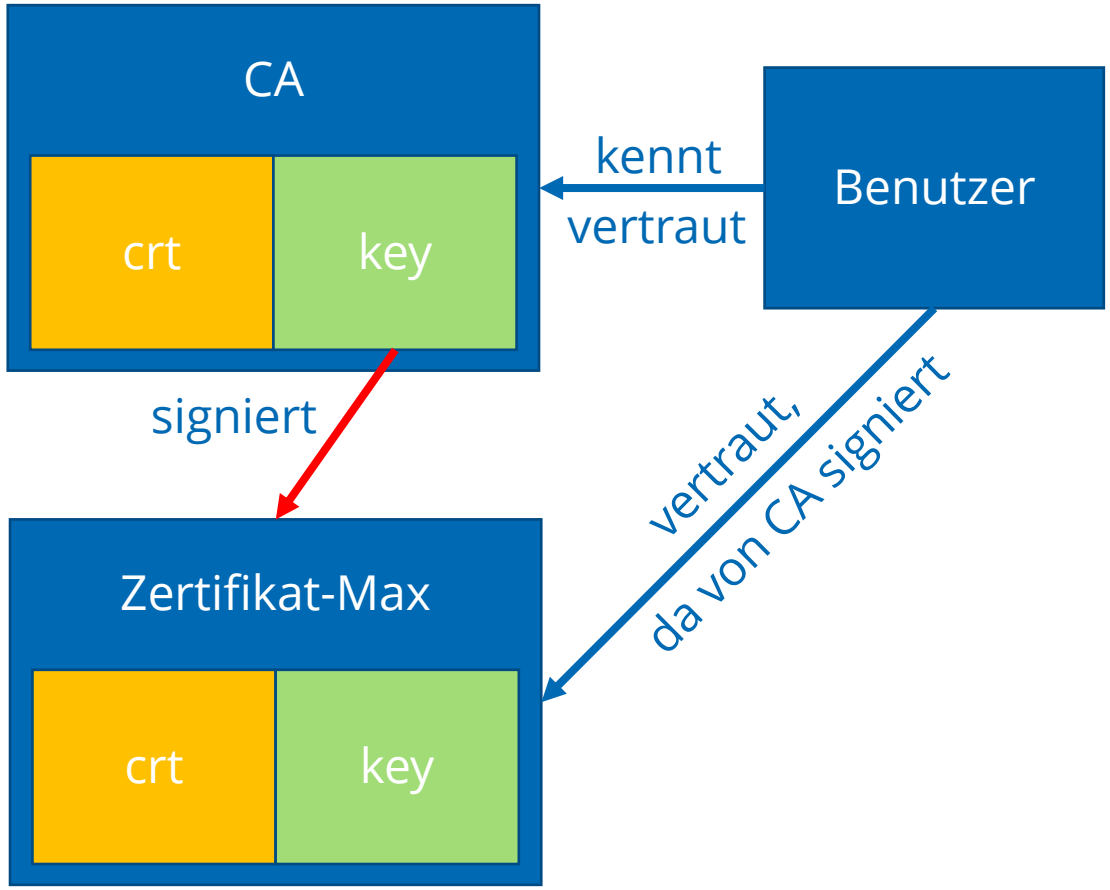
- `sudo iptables -L --line-numbers`
  - Einträge nummeriert anzeigen
- `sudo iptables -D <Ankunftsliste> <Eintragsnummer>`
  - Einträge aus Liste mit Eintragsnummer löschen
- `sudo iptables –F`
  - Alle Einträge löschen

Beispiel:

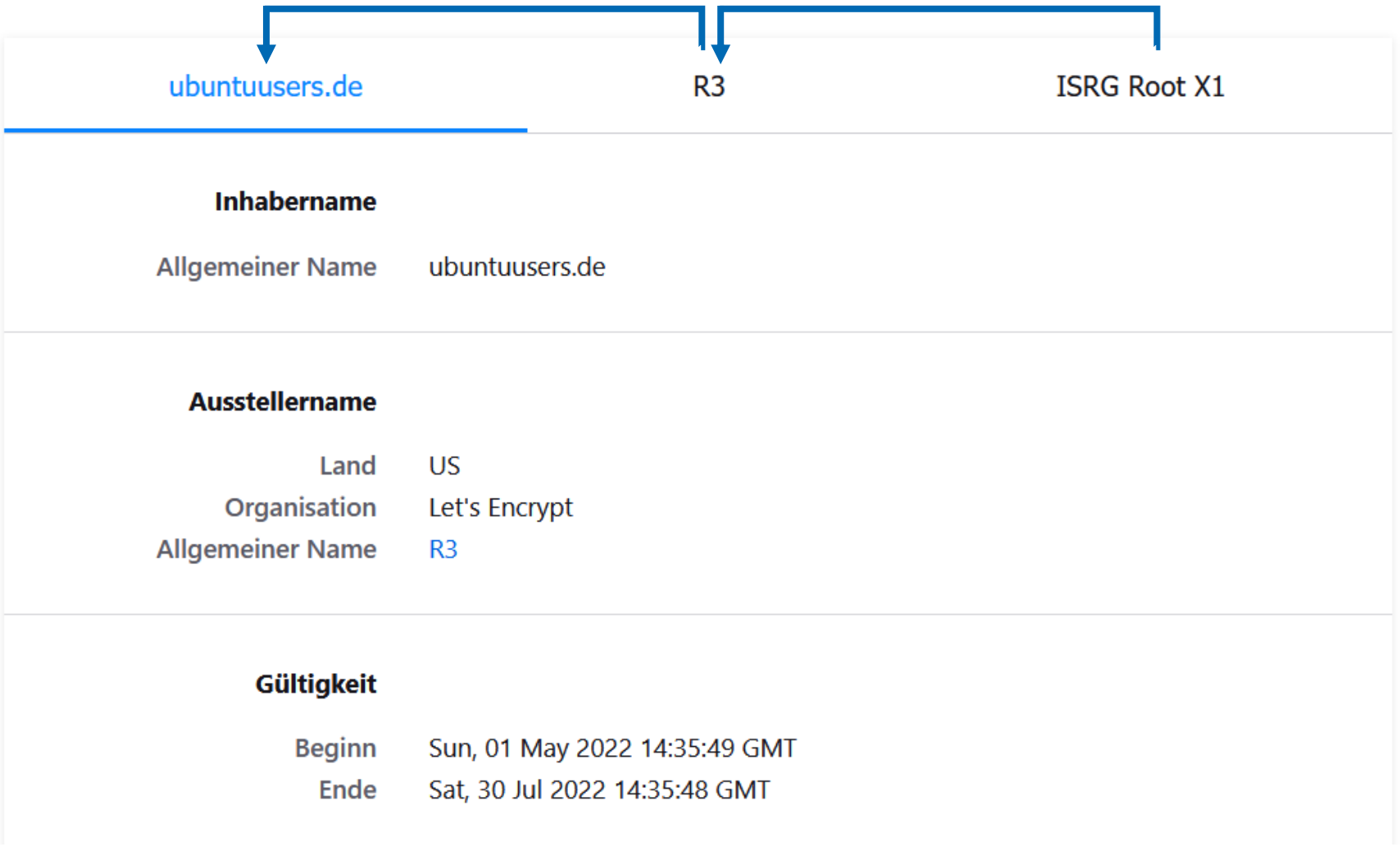
→ **`sudo iptables –D INPUT 1`**

Zertifikate

- X.509 Standard
- Authentifizierung des Servers bei HTTPS
- Starten einer sicheren Verbindung über TLS
- VPN-Authentifizierung
- Signieren
  - PDF-Dateien
  - Programme
  - SSH-Keys
- Oberste Einheit = CA (Certificate Authority)
- .crt = Öffentliches Zertifikat mit öffentlichen Schlüssel
- .key = Privater Schlüssel
- PEM = Speicherformat
- Speicherort: /etc/ca-certificates und /etc/ssl



Zertifikat Beispiel



Zertifikat erstellen

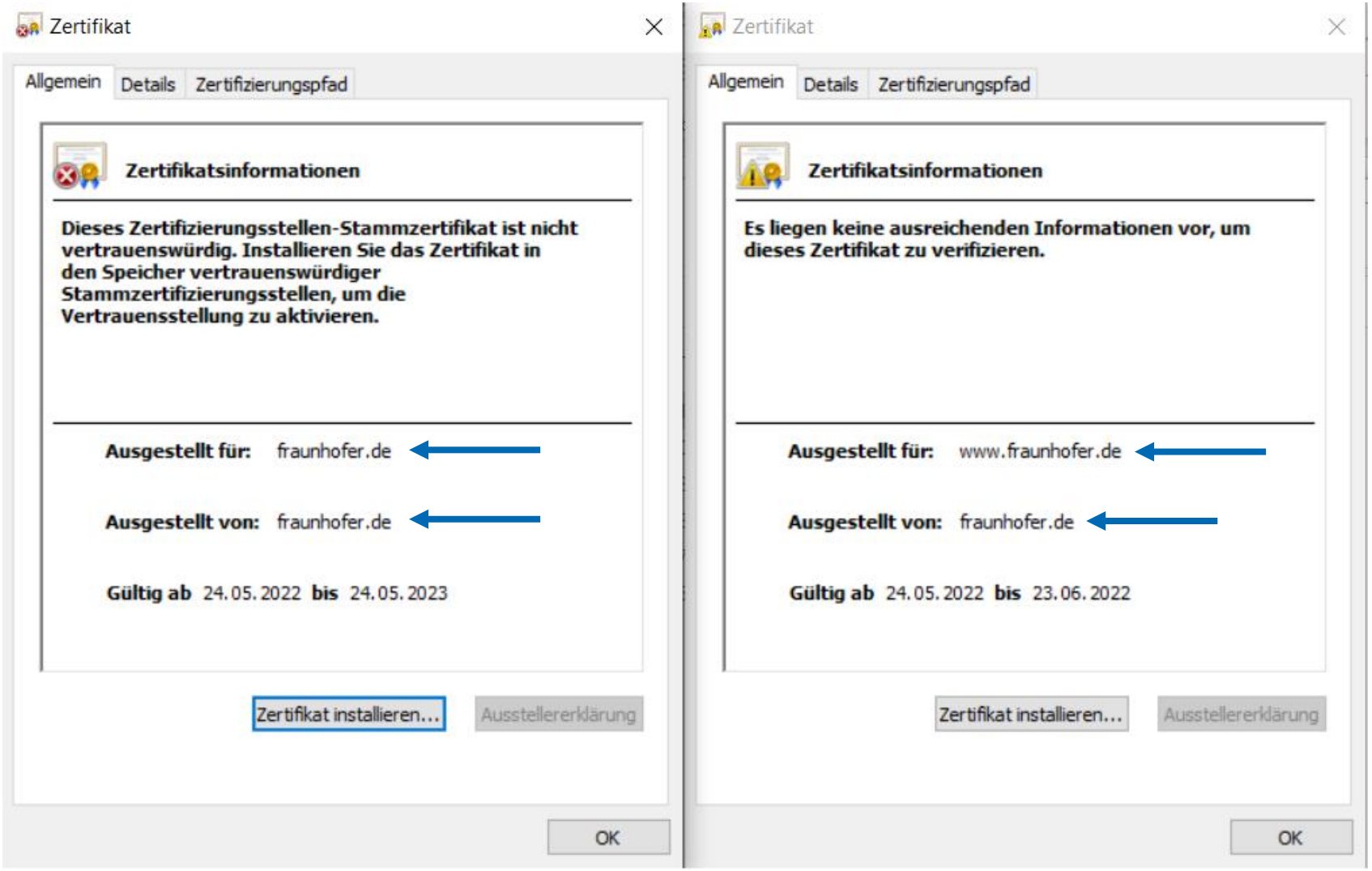
selbstsigniertes Zertifikat erstellen

```
fische11@DESKTOP-5MHG39V:~/ca$ openssl req -newkey rsa:4096 -x509 -sha512 -days 365 -nodes -out ca.crt -
keyout ca.key
Generating a RSA private key
..............................++++
.....................................................................++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:saxony
Locality Name (eg, city) []:Mittweida
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Fraunhofer e.V.
Organizational Unit Name (eg, section) []:Lernlabor Cybersicherheit
Common Name (e.g. server FQDN or YOUR name) []:fraunhofer.de
Email Address []:fische11@hs-mittweida.de
fische11@DESKTOP-5MHG39V:~/ca$
```

Zertifikat erstellen und mit CA signieren

```
fische11@DESKTOP-5MHG39V:~/ca$ openssl genrsa -out webserver.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................................................................
.....................++++
.....................................................................++++
e is 65537 (0x010001)
fische11@DESKTOP-5MHG39V:~/ca$ openssl req -new -key webserver.key -out webserver.csr -extensions v3_req
 -sha512
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:saxony
Locality Name (eg, city) []:Mittweida
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Fraunhofer e.V.
Organizational Unit Name (eg, section) []:Lernlabor Cybersicherheit
Common Name (e.g. server FQDN or YOUR name) []:www.fraunhofer.de
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
fische11@DESKTOP-5MHG39V:~/ca$ openssl x509 -sha512 -req -in webserver.csr -out webserver.crt -CA ca.crt
 -CAkey ca.key -days 30 -CAcreateserial -CAserial ca.seq
Signature ok
subject=C = DE, ST = saxony, L = Mittweida, O = Fraunhofer e.V., OU = Lernlabor Cybersicherheit, CN = ww
w.fraunhofer.de
Getting CA Private Key
fische11@DESKTOP-5MHG39V:~/ca$ ls -l
total 24
-rw-r--r-- 1 fische11 fische11 2236 May 24 17:11 ca.crt
-rw------- 1 fische11 fische11 3272 May 24 17:10 ca.key
-rw-r--r-- 1 fische11 fische11   41 May 24 17:18 ca.seq
-rw-r--r-- 1 fische11 fische11 2061 May 24 17:18 webserver.crt
-rw-r--r-- 1 fische11 fische11 1748 May 24 17:17 webserver.csr
-rw------- 1 fische11 fische11 3243 May 24 17:15 webserver.key
```

Zertifikate erstellen



USB-Sicherheit

- USB-Firewall über **`usbauth`**

- **Sperren möglich von**
  - speziellen Geräten - USB-LTE-Adapter von Telecom nicht erlaubt
  - Gerätegruppen - WLAN-Adapter blockiert
  - festen USB-Ports - Tastatur nur an USB2.0 erlaubt
  - Beschränkung auf bestimmte Anzahl - Nur eine Tastatur erlaubt

USB-Sicherheit

- `allow busnum==3 devpath==2 bInterfaceClass==03 anyChild bInterfaceProtocol==01 devcount<=1`
  - Nur eine Tatstatur an BUS 3, Anschluss 2 Erlaubt

- `allow busnum==2 devpath==6 idVendor==1c7a idProduct==0801`
  - Spezieller Fingerabdrucksensor an BUS 2, Anschluss 6 erlaubt

- `deny all`
  - Alles andere verboten

Rechteverwaltung

Datei-Zugriffsberechtigung

drwxrwxrwx

- TypBesitzerGruppeAlleAnderen
- d: Dateiformat
- r: Leserechte (read)
- w: Schreibrechte (write)
- x: Ausführrechte (execute)
- s: setuid (Ausführung mit Besitzer-rechten)
- s: setgid (Ausführung mit Gruppen-rechten)
- t: stickybit (Programm verbleibt im Speicher nach Ausführungsende)

- Dateiformate
  - -: normale Datei
  - d: Ordner (directory)
  - b: Block-Device
  - c: Zeichen (character)
  - l: Softlink (symbolic link)
  - n: Netzwerk (network)
  - p: Pipe / First-in, First-out
  - s: Socket

Ordnerrechte

- R = Leserechte
  - Auflisten von Dateien im Ordner
- W = Schreibrechte
  - Anlegen, bearbeiten und löschen von Dateien
- X = Zugriff
  - Zugriff auf Dateien im Ordner, wenn Name bekannt

User-Typen

Normal-User

- UID ab 1000
- Benutzerkonto
- Gedacht zur normalen Anmeldung
- Home-Verzeichnis für
  - Persönliche Einstellungen
  - Private Dateien
- Spezieller Nutzer root
  - Vollzugriff, darf Alles
  - UID 0
  - Attraktives Ziel für Angreifer

System-User

- Benutzer für Programme
- Rechteverwaltung von Programmen
  - Programm unter System-User-Kontext starten
  - Benutzer-Rechte bestimmen Programm-Rechte
- Kein Home-Verzeichnis
- Häufig automatisch erstellt bei Programminstallation

Gruppen-Typen

Normale-Gruppe

- Gruppe für Benutzer
- Ab GID 1000
- Root-Gruppe GID 0
- Standardmäßig bei Benutzererstellung
  - Gleichnamige Gruppe
  - gleiche ID

System-Gruppe

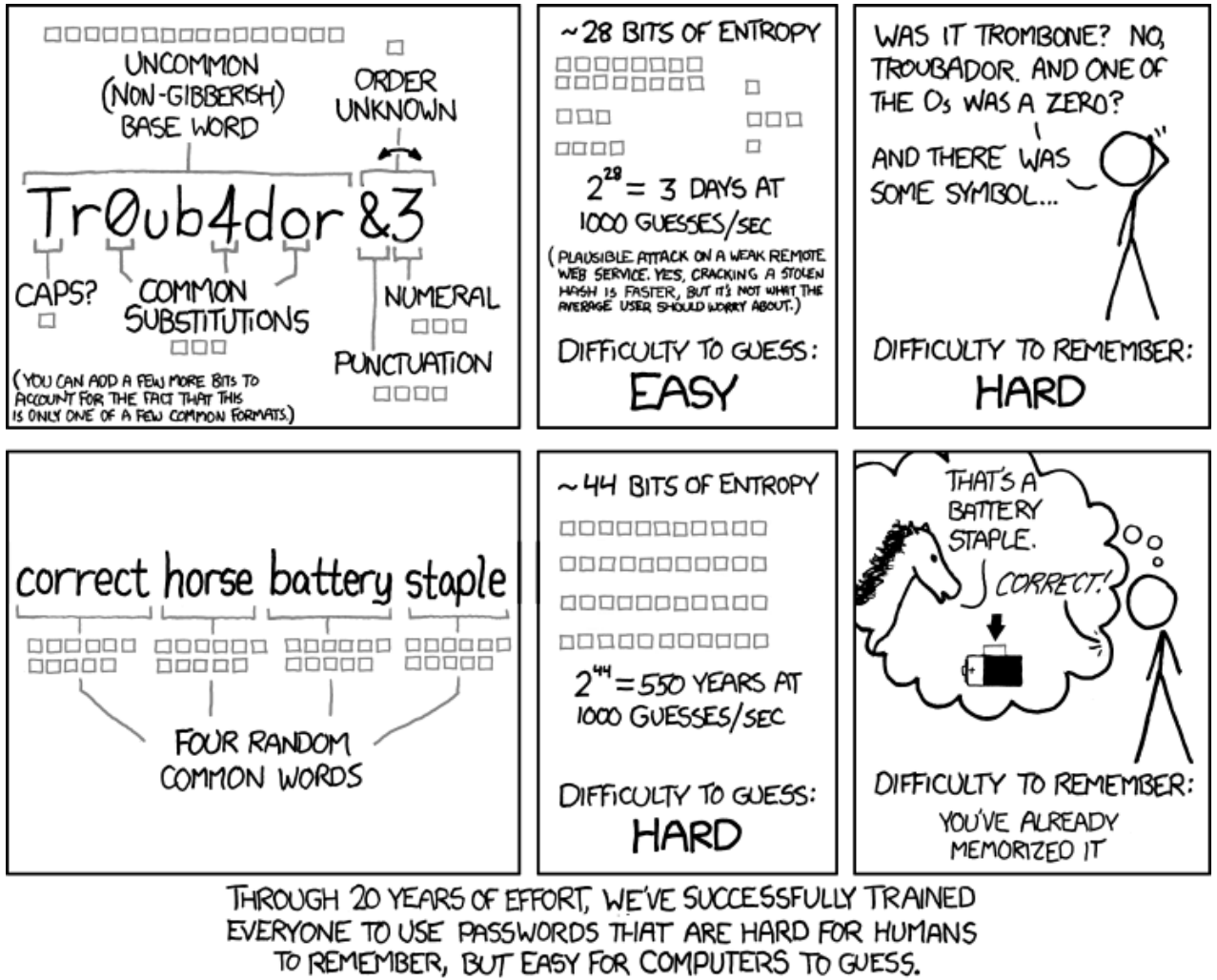
- Gruppe für System-Nutzer
- Kein Unterschied für Rechte zu Normalen Gruppen
- Werden ausgeblendet bei anzeigen von Benutzern (optionale Anzeige)

Passwortsicherheit

- Passwort nicht in Klartext gespeichert
- Passwort-Hash für Passwortabfrage
- Salt als Schutz vor Rainbow-Tables
- Länge schlägt Zeichenanzahl

$$Dauer = \frac{Zeichenanzahl^{Länge}}{Versuche\ pro\ Sekunden}$$

- **Tipp:** 3-4 Wörter und Sonderzeichen / Zahl dazwischen



Quelle: https://xkcd.com/936/

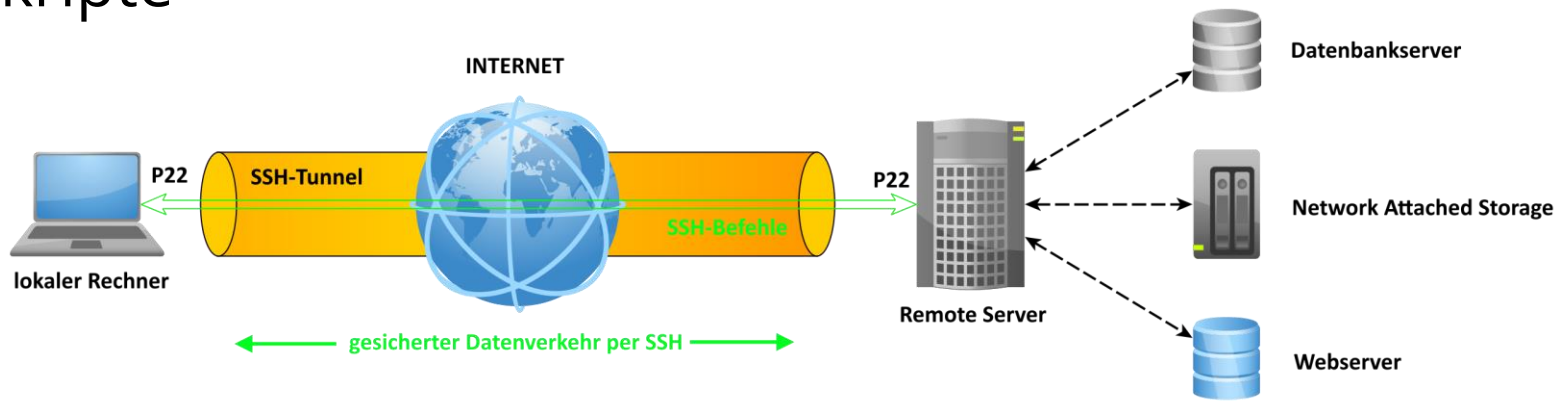
Benutzer sperren

- Eintrag ändern in /etc/shadow
  - Benutzer:*: …
  - !, !! oder * verhindert Login
  - Ersetzt Passwort-Salt und -Hash

Fernzugriff

SSH

- SSH, kurz für Secure Shell
- Sichere Verbindung zu entfernten Computern
  - Verschlüsselt
  - Authentifiziert
- Nutzbar in Automatisierung durch Skripte
- Sichere Dateiübertragung
- Standardmäßig Port 22
- Zusätzlich benutzbar als
  - VPN
  - Graphische Aufschaltung



SSH-Login

- **`ssh benutzer@ip_oder_dns`**
- Bei erster Anmeldung
  - Fingerabdruck des Servers bestätigen
  - Fingerabdruck in ~/.ssh/known_hosts hinterlegt
- Änderung des Fingerabdrucks bedeutet
  - Server neu installiert
  - Man in the Middle Angriff
    (Client ←→ Angreifer ←→ Server)
- -p abweichenden Port zu 22

SSH-Key

- Anmeldung mit Public-Key-Verfahren
- Keine Passworteingabe notwendig
- Verfügbare Kryptographieverfahren
  - RSA (Primfaktoren)
  - DAS (Diskreter Logarithmus)
  - ECDSA und EDDSA (Elliptische Kurven Diskreter Logarithmus)
- Ein Key-Pair pro Ziel-Computer
- Erlaubte Public-Keys in ~/.ssh/authorized_keys (Eine Zeile = Ein Eintrag)
- Eigene Keys in
  - ~/.ssh/key_name (Private Key = Darf Computer **nie** verlassen und unlesbar für andere Benutzer)
  - ~/.ssh/key_name.pub (Public Key = Darf jeder habe)

SSH-Key einrichten

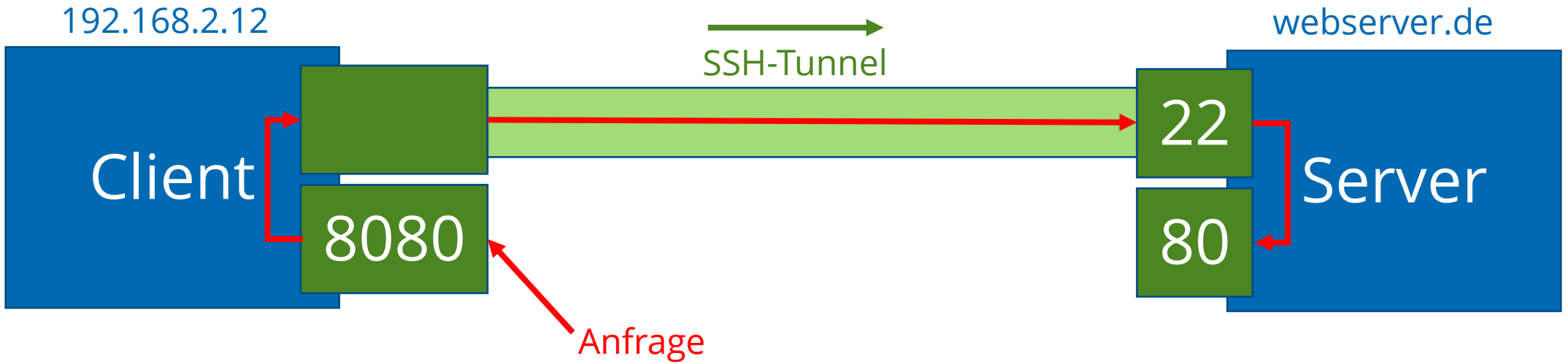
1. Schlüssel erzeugen
   - **`ssh-keygen -t ed25519 -a 420 -f ~/.ssh/server_name -C „Felix Laptop“`**
     - -t Schlüsseltyp ed25519 = Elliptische Kurve
     - -a Hashrunden für privaten Schlüssel (Brute Force Schutz)
     - -f Dateiname
     - -C Kommentar (Hilft zur Identifikation des Schlüssels auf dem Server)
2. Schlüssel auf Server übertragen (alternativ manuell mit scp)
   - **`ssh-copy-id -i ~/.ssh/server_name.pub benutzer@server_ip_oder_dns`**
3. Verbinden mit Nutzen des SSH-Keys
   - **`ssh -i ~/.ssh/server_name.pub benutzer@server_ip_oder_dns`**

SSH-Config

- Konfiguration für SSH-Verbindung
- Speicherort
  - ~/.ssh/config
  - Unter einigen Distributionen auch: ~/.ssh/config.d/*
- Host server_name
  - HostName ssh.example.com
  - User benutzername
  - Port 22
  - PreferredAuthentications publickey
  - IdentityFile ~/.ssh/private_ssh_key
  - IdentitiesOnly yes

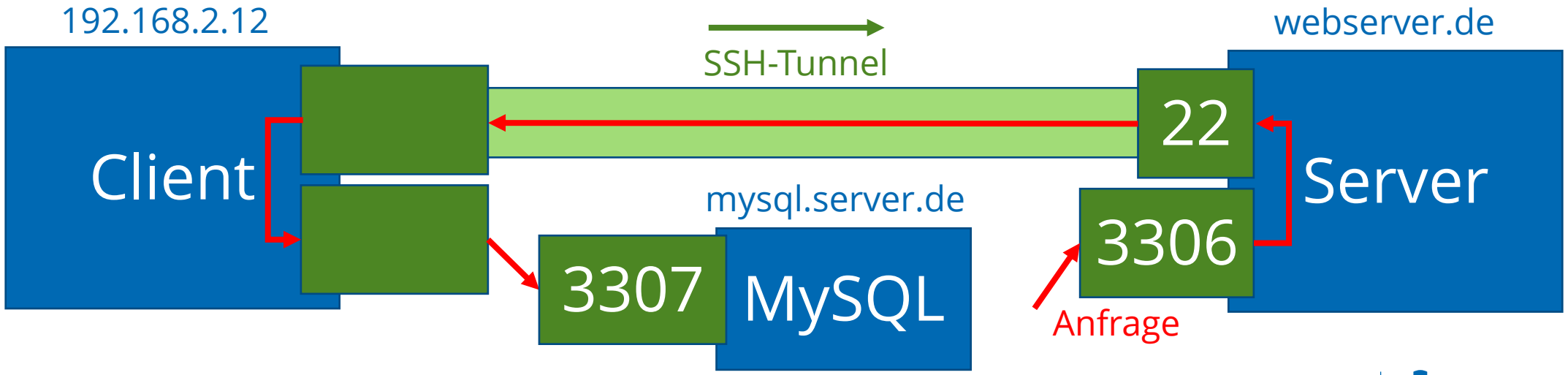
SSH-Tunnel Forward

- `ssh -L <localerPort>:eigeneIP:<serverPort> benutzer@server_ip`
- Beispiel:
  - **`ssh -L 8080:192.168.2.12:80 felix@webserver.de`**
  - SSH-Verbindung mit webserver.de
  - Port 80 vom Server erreichbar unter 192.168.2.12:8080



SSH-Tunnel reverse

- `ssh -R <localerPort>:localeIP:<serverPort> benutzer@server_ip`
- Beispiel:
  - `ssh -R 3307:mysql.server.de:3306 felix@webserver.de`
  - SSH-Verbindung mit webserver.de
  - Port 3307 vom MySQL-Server erreichbar von Webserver unter 3306



SCP

- **scp**, kurz für secure copy
- Kopieren durch SSH-Tunnel
- **scp datei kopie**
  (kein SSH-Tunnel)
- **scp datei benutzer@server:/pfad/dateiname**
- **scp benutzer@server:/pfad/dateiname datei**
- **scp benutzer@server:/pfad/dateiname benutzer@server:/pfad/dateiname**
- Unterstützt Autovervollständigung
- Angabe von Alias möglich, wenn in ~/.ssh/config defniniert

Dateisicherheit

GnuPG (Version2)

- Dateiverschlüsselung
- Unterstützte Algorithmen
  - RSA
  - ECDH
  - ECDSA und EdDSA
  - Elgamal
  - DAS
  - AES
  - Camellia
  - Twofish
  - SHA2 (SHA256, SHA512)

- apt install gnupg2

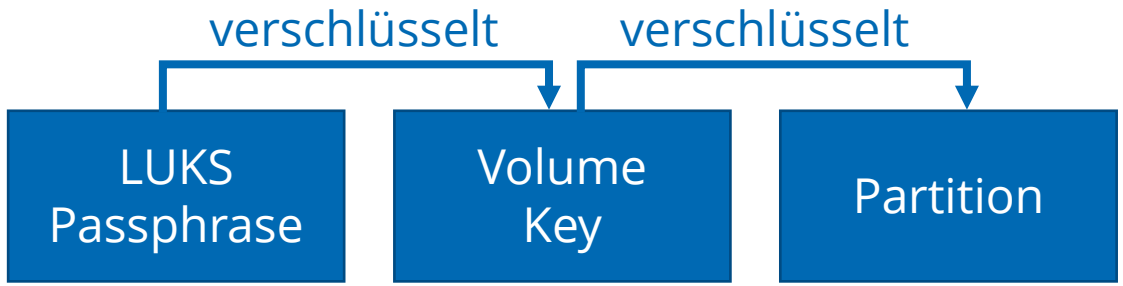
```
fische11@DESKTOP-5MHG39V:~/test$ cat file1
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ gpg2 -c file1
gpg: directory '/home/fische11/.gnupg' created
gpg: keybox '/home/fische11/.gnupg/pubring.kbx' created
fische11@DESKTOP-5MHG39V:~/test$
fische11@DESKTOP-5MHG39V:~/test$ cat file1.gpg
□       □□□54Tv□k^□□    □jj=SF38!□;Z□Z=@YL"<]~)}Yc□('>^oca,;h□4□EK5K□>o□-?□x~Zrfische11@DESKTOP-5MHG39V
~/test$
fische11@DESKTOP-5MHG39V:~/test$ gpg -d -o file1.decrypted file1.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
fische11@DESKTOP-5MHG39V:~/test$ cat file1.decrypted
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ hexdump ~/.gnupg/pubring.kbx
0000000 0000 2000 0101 0200 424b 6658 0000 0000
0000010 8c62 6d9b 8c62 6d9b 0000 0000 0000 0000
0000020
fische11@DESKTOP-5MHG39V:~/test$
```

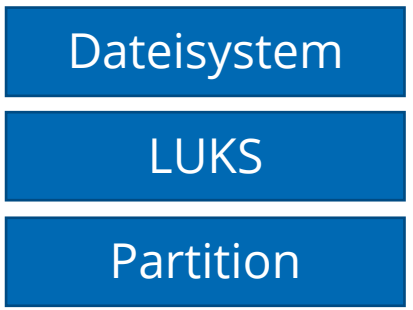
- **Achtung!**
  - Nicht verwechseln mit PGP

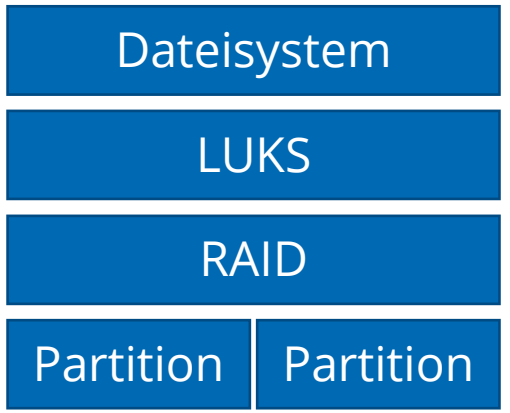
LUKS

- Linux Unified Key Setup
- Festplattenverschlüsselung (Partitionen oder Container)
- LUKS speichert Metadaten über Verschlüsselungsmethode
- Programm cryptsetup
- Automatisiertes Einbinden
- Integriert in Dateibrowsern

```
fische11@DESKTOP-5MHG39V:~/test$ sudo cryptsetup luksFormat --type luks2 /dev/loop1

WARNING!
========
This will overwrite data on /dev/loop1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase for /home/fische11/test/disk1:
Verify passphrase:
WARNING: Locking directory /run/cryptsetup is missing!
fische11@DESKTOP-5MHG39V:~/test$ sudo cryptsetup luksOpen /dev/loop1 loop1-decrypted
Enter passphrase for /home/fische11/test/disk1:
fische11@DESKTOP-5MHG39V:~/test$ sudo mkfs.ext4 /dev/mapper/loop1-decrypted
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 4096 4k blocks and 4096 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

fische11@DESKTOP-5MHG39V:~/test$ mkdir loop1-mount
fische11@DESKTOP-5MHG39V:~/test$ sudo mount /dev/mapper/loop1-decrypted loop1-mount
```







Sicheres Löschen

- **`rm`** entfernt nur Eintrag im Filesystem
  (Radieren im Inhaltsverzeichnis von Buch)
- Daten noch vorhanden, aber als überschreibbar markiert
- Vollständiges Löschen = überschreiben
- Sicheres Löschen = mehrfaches überschreiben
- **`shred -n 5 -uzv /Pfad/Datei`**
  - -n Anzahl an Überschreibungen
  - -u Datei löschen nach überschreiben
  - -z mit Nullen überschreiben beim letzten Durchlauf
  - -v verbose = Fortschrittsinformationen anzeigen
- **`dd if=/dev/urandom of=/dev/Festplatte bs=4k`**

Zusammenfassung

Zusammenfassung

Sie kennen nun Grundprinzipien der IT-Sicherheit.

Unter Linux haben Sie heute Möglichkeiten kennengelernt, wie Sie ein laufendes System auf auffällige Programme untersuchen können. Dazu zählen Programme, wie **`ps`**, **`top`** **und** **`htop`**. Außerdem wissen Sie, wie deren Integrität geprüft werden kann.

Sie können die Netzwerksicherheit eines Systems beurteilen und eigenständig Regeln für dieses festlegen. Sichere Datenübertragung von Befehlen und Daten können Sie mit **SSH** und **SCP** realisieren.

Sie wissen nun den Unterschied zwischen normalen und System-Benutzern bzw. Gruppen. Sie wissen außerdem, wie ein Benutzer gesperrt werden kann.

Heute haben Sie gelernt Daten und Festplatten unter Linux zu verschlüsseln und sicher zu löschen. Sie kennen den Unterschied zwischen einfachen löschen und sicheren löschen.

HOCHSCHULE MITTWEIDA
University of Applied Sciences

Prof. Ronny Bodach

Hochschule Mittweida | University of Applied Sciences
Technikumplatz 17 | 09648 Mittweida
Fakultät Angewandte Computer- und Biowissenschaften

T +49 (0) 3727 58-1011
F +49 (0) 3727 58-21011
@ bodach@hs-mittweida.de
www.cb.hs-mittweida.de

Haus 8 | Richard-Stücklen Bau| Raum 8-205
Am Schwanenteich 6b| 09648 Mittweida

Tim Wetterau B.Sc.

Hochschule Mittweida | University of Applied Sciences
Technikumplatz 17 | 09648 Mittweida
Fakultät Angewandte Computer- und Biowissenschaften

T +49 (0) 3727 58-1752
@ wetterau@hs-mittweida.de

Haus 6 | Grunert de Jacome Bau | Raum 6-031
Am Schwanenteich 4b | 09648 Mittweida

hossfeld@hs-mittweida.de