Sicherer Zugriff auf Daten zu Hause

Zugriff über das Internet auf den PC zu Hause
- OpenVPN und andere sichere Wege -

am LUG-Camp 2010 in Roth
2010-05-13 bis 2010-05-16

Referent:
Bernd Strößenreuther
<lug@stroessenreuther.net>

Lizenz

Anforderung

- Auf dem Rechner zu Hause liegen private Daten in Dateien, eMails, Chat-Protokollen, ...
- früher oder später ergibt sich der Wunsch, von unterwegs darauf zuzugreifen
- das Internet ist dabei ideales Transportmittel, da günstig und fast überall verfügbar
- eine angemessene Absicherung sorgt dafür, dass private Daten auch privat bleiben

Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

Das Problem mit den Passwörtern

- oft kommt für Zugriffsbeschränkungen eine Passwort-Authentifizierung zum Einsatz
- Gute Passwörter (min. 12 Zeichen, Groß- und Kleinbuchstaben, Ziffern, Sonderzeichen bunt gemischt, keine Anlehnung an Wörterbucheinträge) wären für zu Hause hinreichend sicher
- User neigen dazu, einfacher zu merkende Passwörter zu verwenden, die von den verfügbaren Cracker-Tools oft innerhalb von Minuten oder Stunden erraten werden können

Zielstellung

- nur wirklich berechtigte User sollen Zugriff erhalten
- Authentifizierung über kryptographische Schlüssel
- Vorteile
  - deutlich länger als jedes Passwort
  - wirklich zufällig
  - zwei-Faktor-Authentifizierung möglich
    Besitz (Schlüssel) plus Wissen (Passwort)

Drei Tools unter Linux

- SSH / SCP / SFTP
  - für den Kommandozeilen-User
- HTTPS mit Client-Zertifikat
  - auch für reine GUI-User geeignet
- OpenVPN mit Client-Zertifikat
  - der Generalist für fast alle Anwendungsfälle

Voraussetzungen

- Rechner zu Hause muss laufen
- seine Verbindung zum Internet besteht oder kann auf Anforderung (z. B. Anruf beim Rechner) aufgebaut werden
- feste IP-Adresse oder DynDNS-Eintrag
- Port-Forwarding vom DSL-Router zum Rechner

Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

SSH: Für wen?

- Kommandozeilen-User
- Konsole vom Rechner zu Hause
- Übertragung von Files per scp / sftp
- mit SSH-Tunnel auch geschützte Übertragung von anderen Protokollen

SSH: Server konfigurieren

- `ssh-keygen -t rsa`
- erzeugt `~/.ssh/id_rsa` (privater Schlüssel) und `~/.ssh/id_rsa.pub` (öffentl. Schlüssel)
- `id_rsa.pub` am Server nach `~/.ssh/authorized_keys`
- `chmod 600 authorized_keys`
- in `/etc/ssh/sshd_config` eintragen:
  ```
  PasswordAuthentication no
  PermitRootLogin no
  ```
- `/etc/init.d/sshd restart`

SSH: Client unter Linux

- `id_rsa` mitnehmen (USB-Stick)
- Client unter Linux:
  `ssh -i /media/id_rsa ↷`
  `booboo@dunno.dyndns.info`
- `scp -i /media/id_rsa ↷`
  `booboo@dunno.dyndns.info:/tmp/↷`
  `meine-datei.txt .`
- oder `scp -i /media/id_rsa datei.txt ↷`
  `booboo@dunno.dyndns.info:/tmp/`
- X-Forwarding meist problemlos

SSH: Client unter Windows

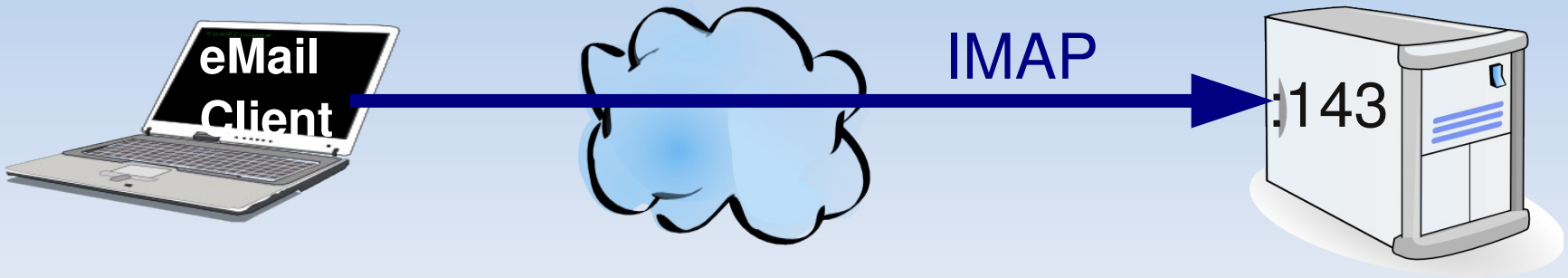
- Putty portable mit auf den Stick
- `id_rsa` konvertieren mit Putty Key Generator `puttygen.exe` -> Conversions -> Import key
- Save private key, Dateiendung `.ppk`
- Diese Datei auf den Stick
- `e:\putty\pscp -i ↷`
  `e:\keys\mein_ssh_key.ppk ↷`
  `booboo@dunno.dyndns.info:/tmp/↷`
  `meine-datei.txt .`
- Meist kein X-Server vorhanden

SSH Tunnel

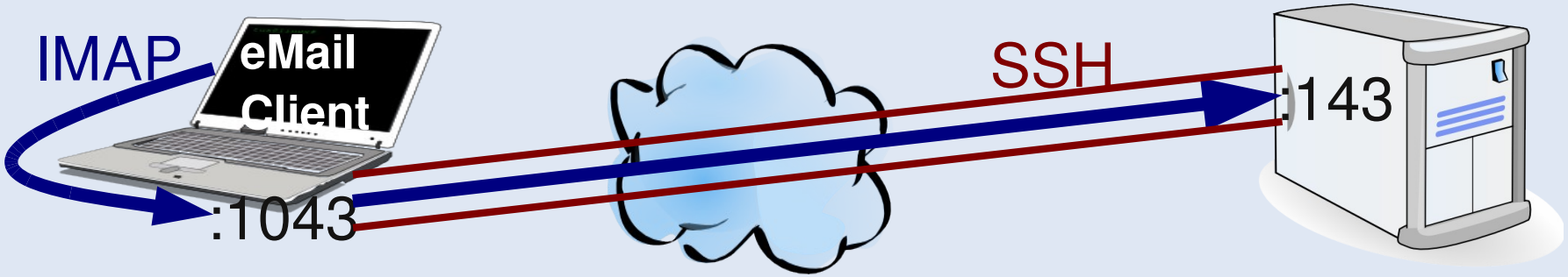
- funktioniert wie ein Verlängerungskabel
- Daten werden mit durch die verschlüsselte SSH-Verbindung geschleust

SSH-Tunnel: Skizze

Direkte Verbindung



SSH-Tunnel



SSH-Tunnel aufbauen

- `ssh -i /media/id_rsa -L ↷ 1043:127.0.0.1:143 ↷ booboo@dunno.dyndns.info`

- Nachteile:
  - nur einzelne Ports
  - Client-Software muss umkonfiguriert werden
  - bei HTTP funktionieren Links oder Redirects mit Domainnamen nicht mehr
  - eher für fachkundige User

SSH-VPN

- Routing zwischen Client und Server über ein separates Interface (z. B. tun0) und ein zu definierendes Transfer-Netz (Point-to-Point)
- ssh Option -w
- man ssh, Abschnitt ”SSH-BASED VIRTUAL PRIVATE NETWORKS”
  auch unter: http://www.openbsd.org/cgi-bin/man.cgi?query=ssh&sektion=1#SSH-BASED+VIRTUAL
- HowTo: http://www.debian-administration.org/articles/539

Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

Exkurs: Zertifikate erzeugen

- OpenVPN und HTTPs arbeiten mit Zertifikaten nach X.509
- Wir erzeugen uns 3 Zertifikate:
  - CA-Zertifikat, das alle weiteren signiert
  - Server-Zertifikat
  - Client-Zertifikat

Umgebung vorbereiten

```
[booboo@dunno ~]$ umask 077
[booboo@dunno ~]$ mkdir ca
[booboo@dunno ~]$ cd ca
[booboo@dunno ca]$ mkdir newcerts
[booboo@dunno ca]$ echo -ne "01" >serial
[booboo@dunno ca]$ echo -ne "01" >crlnumber
[booboo@dunno ca]$ locate openssl.cnf
[booboo@dunno ca]$ cp <pfad>/openssl.cnf ↷
 ./ca.cnf
```

ca.cnf anpassen

- in ca.cnf werden individuelle Vorgaben konfiguriert
- im Abschnitt `[ CA_default ]`

```
dir             = /home/booboo/ca
certificate = $dir/ca.example.com.crt
private_key = $dir/ca.example.com.key
```

CA-Zertifikat erzeugen

```
[dunno ca]$ openssl genrsa -des3 -out ↷
ca.example.com.key 2048
[...]
Enter pass phrase for ca.example.com.key:********
Verifying - Enter pass phrase for ↷
ca.example.com.key:********
[dunno ca]$ openssl req -config ./ca.cnf -new ↷
-x509 -days 3650 -key ca.example.com.key ↷
-out ca.example.com.crt
Enter pass phrase for ca.example.com.key:********
[...]
Country Name (2 letter code) [GB]:DE
State or Province Name (full name) [Berkshire]:Bayern
Locality Name (eg, city) [Newbury]:Nuernberg
Organization Name (eg, company) [My Company Ltd]:BooBoo
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your servers hostname) []:↷
ca.example.com
Email Address []:ca@example.com
```

Server-Zertifikat erzeugen

```
[dunno ca]$ openssl genrsa -out dunno.dyndns.info.key 1024
[...]
[dunno ca]$ openssl req -config ./ca.cnf -new ↷
-key dunno.dyndns.info.key -out dunno.dyndns.info.csr
[...]
Common Name (eg, your name or your servers hostname) []:↷
dunno.dyndns.info
Email Address []:webmaster@example.com
[...]
[dunno ca]$ openssl ca -config ./ca.cnf -days 730 ↷
-in dunno.dyndns.info.csr -out dunno.dyndns.info.crt
Using configuration from ./ca.cnf
Enter pass phrase for ca.example.com.key:********
Check that the request matches the signature
Signature ok
Certificate Details: [...]
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
```

Client-Zertifikat erzeugen

```
[dunno ca]$ openssl genrsa -des3 ↷
-out hans.meier.example.com.key 1024
[...]
[dunno ca]$ openssl req -config ./ca.cnf -new ↷
-key hans.meier.example.com.key ↷
-out hans.meier.example.com.csr
[...]
Common Name (eg, your name or your servers hostname) []:↷
hans.meier.example.com
Email Address []:hans.meier@example.com
[...]
[dunno ca]$ openssl ca -config ./ca.cnf -days 730 ↷
-in hans.meier.example.com.csr ↷
-out hans.meier.example.com.crt
[...]
```

Client-Zertifikat in PKCS12

- Client-Zertifikat und -Key (privater Schlüssel) werden in eine gemeinsame Datei im Format PKCS12 geschrieben, die gängige Browser importieren können

```
[dunno ca]$ openssl pkcs12 -export ↷
-in hans.meier.example.com.crt ↷
-inkey hans.meier.example.com.key ↷
-certfile ca.example.com.crt ↷
-out hans.meier.example.com.p12
Enter pass phrase for ↷
hans.meier.example.com.key:********
Enter Export Password:********
Verifying - Enter Export Password:********
```

Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

HTTPS: Für wen?

- auch für reine GUI-Anwender
- gewohnte Oberfläche: Browser
- Variante mit Web-Oberfläche ist von vielen Anwendungen verfügbar
  (Web-Mailer, Datenbank-Frontends, Video-DB, Konsolen, ...)
  müssen am Server installiert werden
- Download von Files standardmäßig möglich
- Upload per WebDAV

Apache konfigurieren

- Ausgangspunkt: VirtualHost im Apache, SSL-enabled, nur Server-Zertifikat

```
<VirtualHost _default_:443>
   DocumentRoot "/var/www/html"
   ServerName dunno.dyndns.info
   ServerAdmin himself@example.com
   ErrorLog logs/error_log
   TransferLog logs/access_log
   SSLEngine on
   SSLCipherSuite HIGH:MEDIUM
   SSLCertificateFile ↷
/etc/httpd/conf/ssl.crt/dunno.dyndns.info.crt
   SSLCertificateKeyFile ↷
/etc/httpd/conf/ssl.key/dunno.dyndns.info.key
</VirtualHost>
```

ein Verzeichnis freigeben

- Verzeichnis soll per HTTPS erreichbar sein
- Directory-Browsing wird erlaubt

```
Alias /data/ "/opt/data/"
<Location /data>
    Options Indexes
</Location>
```

- Freigabe von Home-Verzeichnissen:
  mod_userdir
  Beispiel siehe httpd.conf

Browser vorbereiten

- am besten einen Firefox portable auf eine Stick installieren
- der Browser muss unserer CA vertrauen, daher CA-Zertifikat einbinden, unter Firefox 3 z. B. Extras -> Einstellungen... -> Erweitert -> Verschlüsselung -> Zertifikate anzeigen -> Zertifizierungsstellen -> Importieren
- Der erste Zugriff sollte jetzt funktionieren, ohne Zertifikatswarnung

Client-Zertifikat erforderlich

- nur Clients, die ein entsprechendes Client-Zertifikat vorweisen können, sollen Zugriff erhalten
- innerhalb des VirtualHost Abschnitts:

```
SSLCACertificateFile /etc/httpd/conf/↷
ssl.crt/ca.example.com.crt
<Location />
    Order allow,deny
    Allow from all
    SSLVerifyClient require
</Location>
```

Browser konfigurieren

- Zugriff sollte jetzt abgewiesen werden
- Client-Zertifikat (*.p12) importieren
- im Zertifkatsmanager unter
  Ihre Zertifikate -> Importieren
- Passwortschutz des Zertifikats wird **nicht** übernommen!!
- daher: Extras -> Einstellungen... -> Sicherheit:
  Haken setzen bei ”Master-Passwort verwenden”
  anschliessend: ”Master-Passwort ändern...”
- ggf. Startseite setzen: https://dunno.dyndns.info/

Portabler Browser unter Linux

- http://stadt-bremerhaven.de/2008/10/24/portable-firefox-303-englisch-fr-linux-testversion/
- https://www.privacyfoundation.de/wiki/PortableLinuxApps

HTTPS: Zusammenfassung

- Sehr einfach für den Anwender
  - Internet-PC suchen
  - Stick anstecken
  - Doppelklick auf FirefoxPortable.exe
  - Frage nach Master-Passwort richtig beantworten
  - Drin!
- Beliebige Browser-Anwendungen und Info-Seiten
- Stark erweiterbar, z. B. zusätzlich Username und Passwort abfragen, Client-IPs einschränken, ...

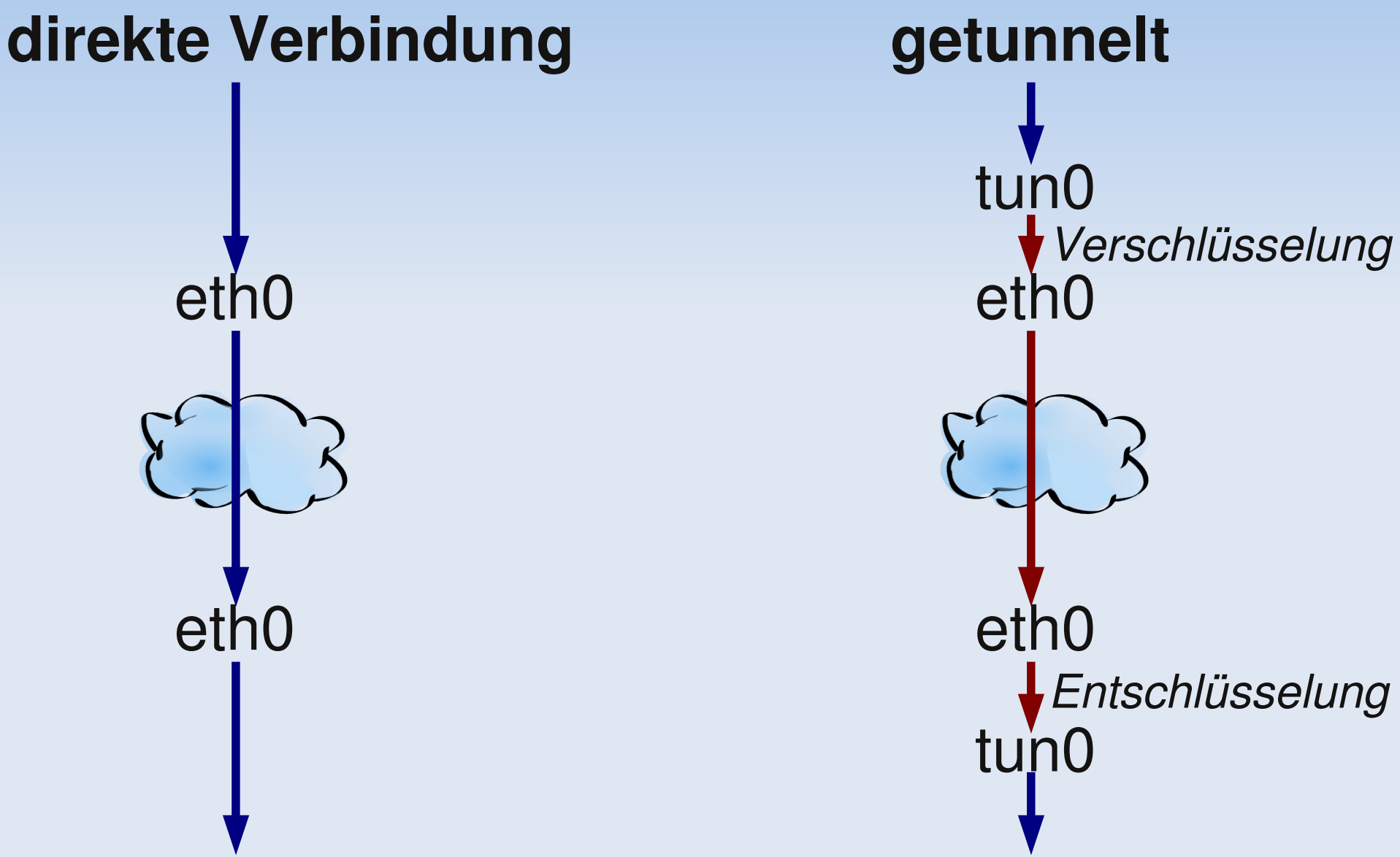
Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

OpenVPN: Für wen?

- vollwertiges VPN
- für Leute, die Ihr Notebook immer dabei haben
- nicht als PortableApp möglich, da tun-Interface
- sowohl für GUI-Anwender als auch Kommandozeilen-User geeignet
- Clients müssen bei Wechsel zwischen LAN und VPN nicht umkonfiguriert werden (gleichbleibende IPs und Ports)
- ebenfalls für Linux und Windows verfügbar

Exkurs: tun-Interface



OpenVPN Server konfigurieren (1)

/etc/openvpn/server.conf Teil 1:

```
# alle Pfade sind relativ zu diesem Verzeichnis
cd /etc/openvpn
auth SHA1
# nur Zertifikate erlauben, deren CN passt
tls-remote hans.meier.example.com
# IP-Adresse, Port und Protokoll fuer Listener
local 192.168.0.3
port 1194
proto udp
# Interface
dev tun0
# CA-Zertifikat
ca ca.meine-domain.crt
```

OpenVPN Server konfigurieren (2)

/etc/openvpn/server.conf Teil 2:

```
# Server-Zertifikat und -Key
cert dunno.dyndns.info.crt
key key/dunno.dyndns.info.key
# Diffie hellman parameter
dh dh1024.pem
# VPN-Subnetz fuer Client-IP-Adressen:
# ein freies Subnetz verwenden
server 192.168.29.0 255.255.255.0
# Zuordnung Client zu IP wird hier gespeichert
ifconfig-pool-persist ipp.txt
# push: Optionen, die dem Client beim
# Verbindungsaufbau mitgegeben werden
# Default-Gateway auf das VPN setzen
push "redirect-gateway"
```

OpenVPN Server konfigurieren (3)

/etc/openvpn/server.conf Teil 3:

```
# zu verwendender DNS-Server, falls im Heim-Netz
# vorhanden
push "dhcp-option DNS 192.168.27.3"
# Praefix, fuer Hostnames ohne Domain
push "dhcp-option DOMAIN home.example.com"
# Keep alive pings ueber VPN schicken
keepalive 10 120
# Daten-Kompression im VPN einschalten
comp-lzo
# Maximale Anzahl gleichzeitiger Clients
max-clients 10
# Status-File
status openvpn-status.log
# Log-Level (0-9)
verb 3
```

OpenVPN-Server starten

```
/etc/init.d/openvpn start
```

OpenVPN-Client konfigurieren (1)

/etc/openvpn/dunno.dyndns.info.c2n.conf Teil 1:

```
cd /etc/openvpn
remote dunno.dyndns.info
port 1194
proto udp
dev tun0
# Server darf Einstellungen per push setzen
pull
# Configuration fuer Client
tls-client
auth SHA1
ca ca.example.com.crt
cert hans.meier.example.com.crt
key key/hans.meier.example.com.key
comp-lzo yes
```

OpenVPN-Client konfigurieren (2)

/etc/openvpn/dunno.dyndns.info.c2n.conf Teil 2:

```
verb 3
# spez. Script von Ubuntu, um resolv.conf bei
# Verbindungsauf- und -abbau dynamisch anzupassen
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
tls-remote dunno.dyndns.info
```

OpenVPN-Client starten

```
/usr/sbin/openvpn --config ↷
/etc/openvpn/dunno.dyndns.info.c2n.conf
```

- Terminal-Fenster offen lassen, solange man die VPN-Verbindung benötigt, anschliessend [CRTL]+[c] um die Verbindung zu beenden
- Alternativ über die Network-Manager der einzelnen Desktops oder Distributionen

OpenVPN weitere Möglichkeiten

- mit ”`redirect-gateway`” entspannt surfen in öffentlichen WLANs: Mitlesen durch andere User wird effizient unterbunden
- OpenVPN ist deutlich mächtiger, als das kleine Beispiel zeigen kann
- ggf. zusätzlich Username/Passwort abfragen
- siehe z. B. http://openvpn.net/index.php/open-source/documentation/howto.html

Agenda

- Zielstellung
- SSH mit Key-Authentifizierung
  - SSH-Tunnel
- Exkurs: Zertifikate erstellen
- HTTPS mit Client-Zertifikat
- OpenVPN mit Client-Zertifikat
- Light-Version: VPN-Endpunkt am DSL-Router
- Zusammenfassung

VPN mit DSL-Router (1)

- Light-Version mit Einschränkungen bei der Sicherheit
- meist nur PPTP als gangbarer Weg
  - nur Authentifizierung mit Username/Passwort
  - Probleme mit Verbindungsabbrüchen
  - GRE hat teilweise Probleme mit NAT
  - inkompatible ”Varianten” von PPTP im Umlauf

VPN mit DSL-Router (2)

- IPsec
  - deutlich komplexer als die bisher gezeigten Wege
  - damit fehleranfällig
  - Probleme mit NAT
- OpenVPN nur mit OpenWRT
  - http://openwrt.org/
  - http://de.wikipedia.org/wiki/OpenWRT

Knoppix auf USB-Stick

- alle Zugangswege lassen sich auch von einem Knoppix verwenden, das man auf einem USB-Stick dabei hat
- Nachteile:
  - ggf. Probleme beim Booten von USB
  - Internet-Zugang muss jeweils separat konfiguriert werden
    - Bei Ethernet-Anbindung an DSL-Router problemlos
    - ISDN, WLAN, reines DSL-Modem, ... ggf. aufwändig

Zusammenfassung

- sicherer Zugriff auf Linux ist über verschiedenste Wege möglich
- auch Windows-Clients können problemlos angebunden werden
- Auswahl des Zugriffsweges nach Einsatzzweck
- möglichst keine zusätzlichen (schlechter gesicherten) Dienste nach aussen hin anbieten

Weiterführende Literatur

- Tiny-CA: http://tinyca.sm-zone.net/
- http://www.admin-magazin.de/content/tiny-ca-als-zertifzierungsstelle?category=370
- für Einsteiger beim Thema OpenVPN:
  ”OpenVPN - Das Praxisbuch”
  Dirk Becker
  Galileo Computing
- VPN-Endpunkt am DSL-Router:
  c't 5/2010 Seite 148 ff

Vielen Dank...

... für die Aufmerksamkeit

Noch Fragen?

- hier und jetzt
- hier und später
- jederzeit lug@stroessenreuther.net