

# Connectivity Everywhere



Ich bin im Netz, aber wie komme ich sicher nach hause?

Tricks fuer mobile  
Internet Nutzer



# Überblick



- Sicherheitsprobleme beim mobilen IP-Nutzer
- Konventionelle Lösung: IP-Tunnel (Layer 3)
- Die Alternative: TCP und andere Tunnel (L4 +)
- SSL & stunnel
- SSH – viele Kanäle
- Beispiele:
  - Mail senden via SMTP (SSH)
  - Zugriff auf Firmenserver (SSH & stunnel)
- Zusammenfassung

# Mobiler Nutzer vs. Sicherheit



## ■ Der mobile Nutzer

- Zugriff auf das Internet von verschiedenen Orten
- Zugriff von verschiedenen Maschinen und Betriebssystemen
- Zugriff von fremden Rechnern aus
- Zugriff aus nicht gesicherten und nicht vertrauenswürdigen Netzen heraus

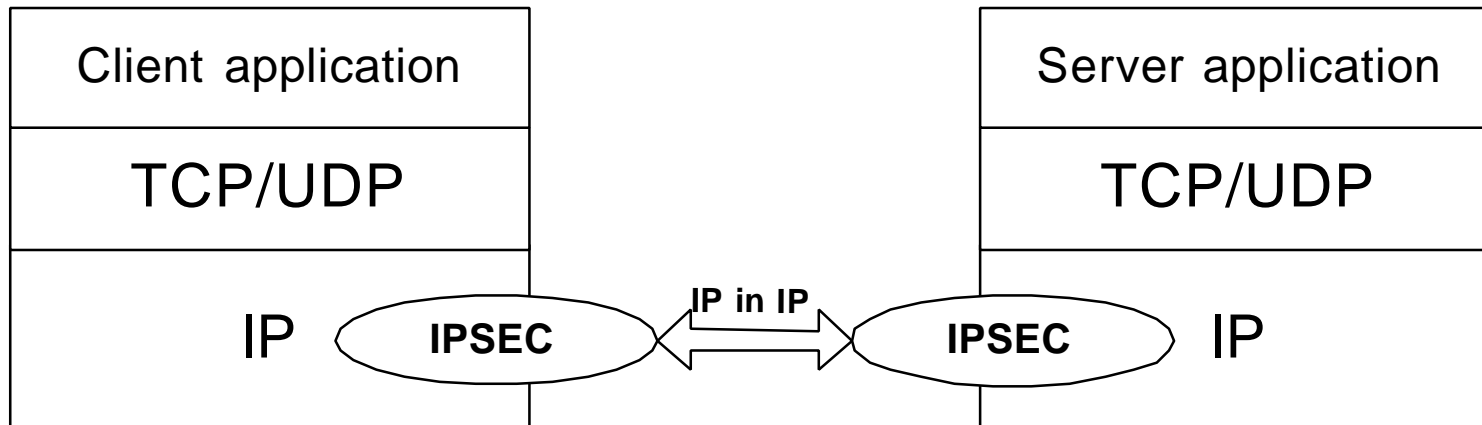
## ■ Und die Sicherheit

- Meist werden nur einige Dienste unterwegs genutzt, die sind aber sicherheitsrelevant (z.B. private oder Firmen-Email)
- “Sniffer”, die gezielt nach Klartext-Passwörtern in beliebten Diensten (z.B. POP & FTP) suchen, sind allgegenwärtig

# IP Security – IPSec u.a.



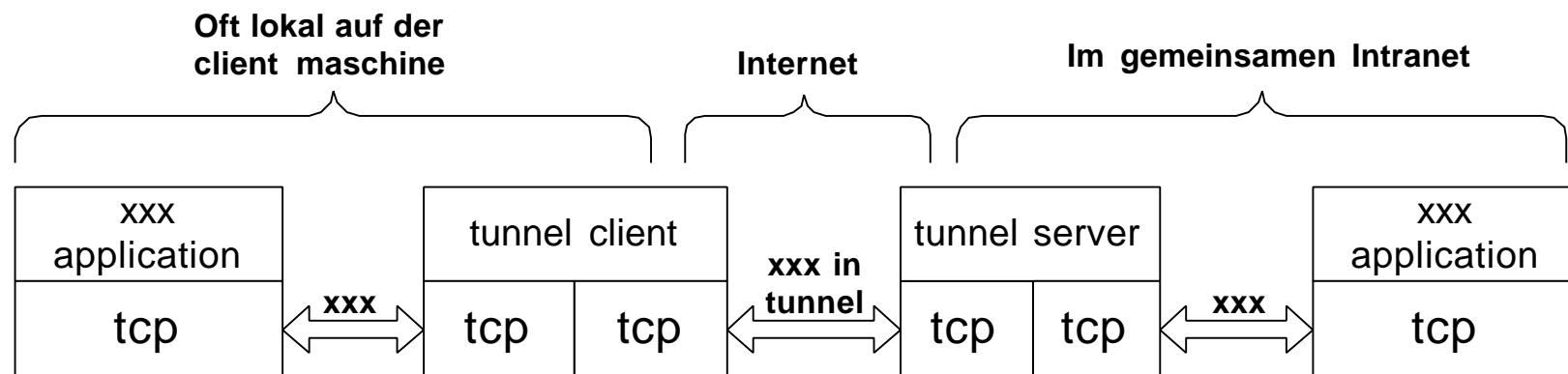
- IP-Pakete werden verschlüsselt und wieder in IP eingepackt
- Erfordert System-Software auf dem Client und dem Server (oder dessen Gateway)
- Häufig Probleme mit NAT/Masq und Firewalls



# Anwendungs-Tunnel Überblick



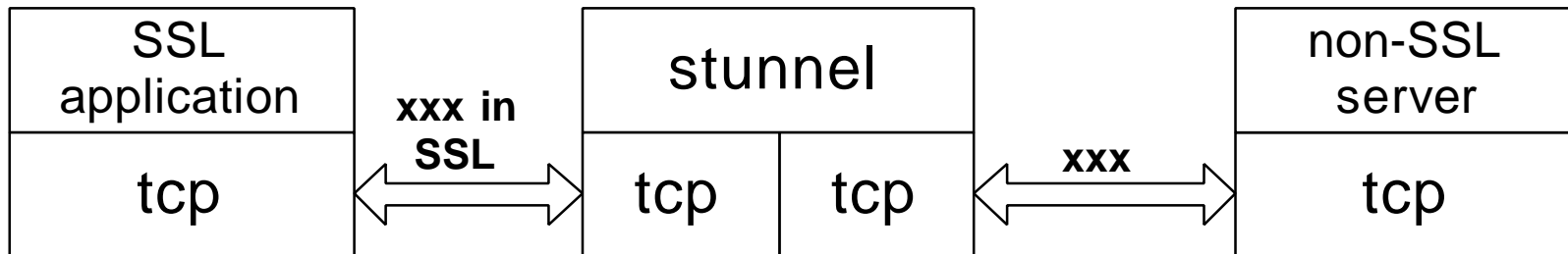
- Streams werden von normalen Anwendungen verschlüsselt – keine Systemsoftware
- Viele verschiedene Varianten von Tunneln sind verfügbar: stunnel und ssh setzen auf TCP auf. Andere nutzen HTTP oder ICMP-echo/reply (ping) pakete



# SSL & stunnel



- Stunnel nimmt SSL-Verbindungen an, kann sich authentisieren und öffnet eine TCP-Verbindung zu einem Server auf der die Daten aus der SSL-Verbindung in Klartext laufen

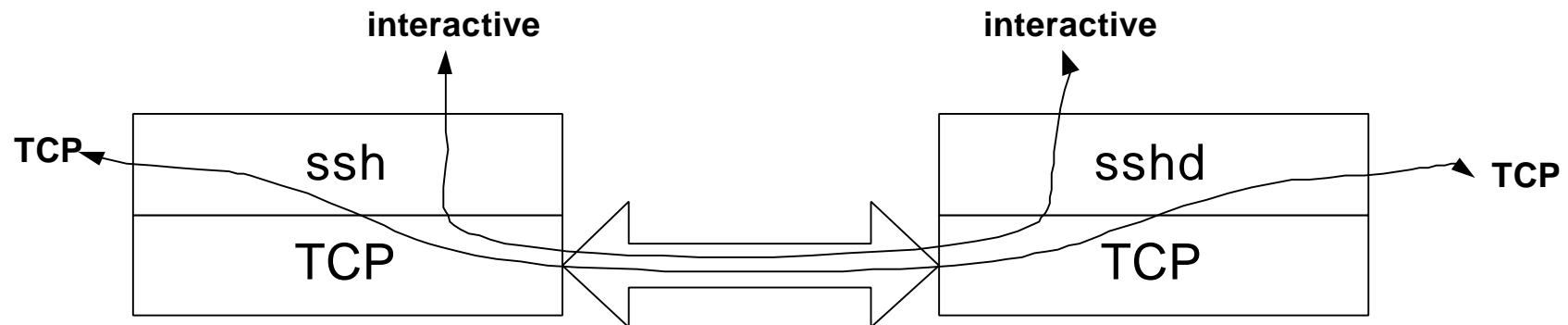


- Beispiele: Frontend zu POP- und IMAP-server im KNF
- Potentielles Problem: Der Server kennt nicht die wahre IP-Adresse des clients

# SSH – viele Kanäle!



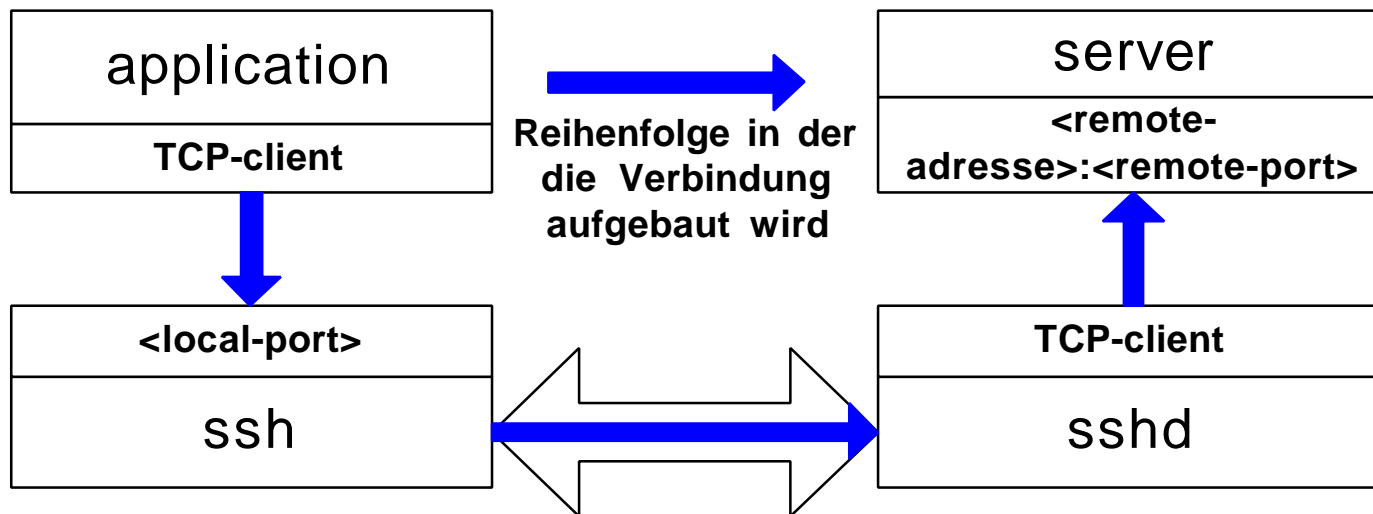
- Ähnliche Arbeitsweise wie SSL aber aber fuer interaktive Nutzung (SSH = „secure shell“) konzipiert:
  - Mehrere Kanäle können in einer SSH-Verbindung gemultiplext werden
  - Mehr interaktive Authentisierungsoptionen
- Verfügbar fuer (fast) alles was eine CPU hat: z.B. Unix, Windows (OpenSSH & TeraTerm) und Mac und Java



# SSH, local port forwarding



- Der ssh-client nimmt auf einem konfigurierbaren TCP-port Verbindungen entgegen und leitet die Daten in einem SSH-Kanal an den Server
- Der SSH-Server öffnet für so einen Kanal eine TCP-Verbindung zu einer konfigurierbaren Adresse/Port und leitet alle Daten weiter
- Parameter: -l <local-port>:<remote-adresse>:<remote-port>

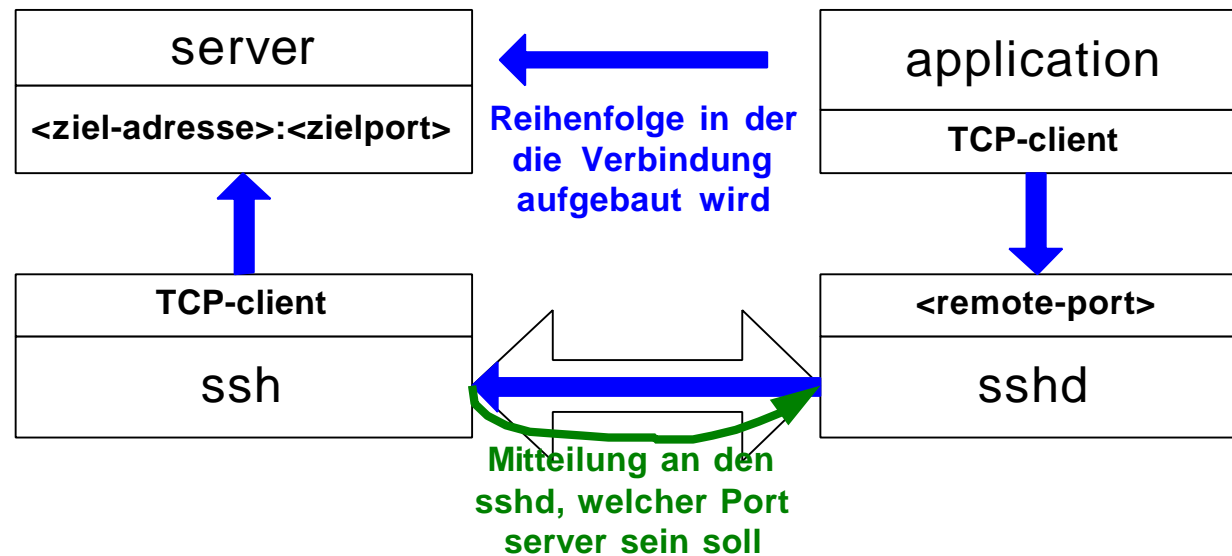




# SSH, remote port forwarding



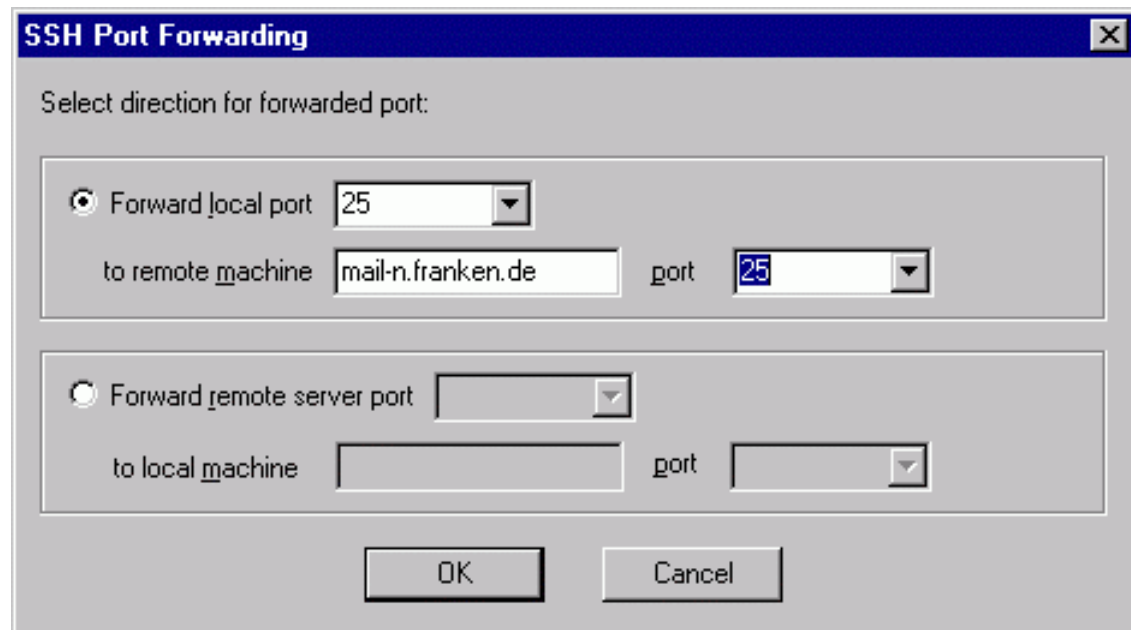
- SSH-Server wird vom Client gebeten auf einem bestimmten TCP-Port zu lauschen. Verbindungen die auf diesem Port angekommen werden angenommen und in einem verschlüsselten Kanal an den client weitergereicht
- Der Client öffnet fuer jede beim SSH-Server eingehende Verbindung eine korrespondierende TCP-Verbindung zum endgültigen Ziel
- Parameter: `-r <remote-port>:<ziel-adresse>:<ziel-port> [-g]`



# Beispiel: Mail losschicken



- SSH-Verbindung zu einem KNF-Rechner
- Den lokalen TCP-port 25 (smtp) auf den Mailserver smtp-port weiterleiten
- Und als Mailserver ‚localhost‘ eingeben



# MindTerm – SSH in Java



- Mit MindTerm kann man SSH auf jedem Rechner mit Java-fähigem Browser nutzen!

The image shows a screenshot of the MindTerm application. The main window is a terminal window titled 'mnbokaem@shell.franken.de <SSH-1.99-OpenSSH\_2.2.0p1> [80x24]'. It displays the following text:

```
SSH Server/Alias: shell.franken.de
No settings file for shell.franken.de found.
Save as alias: shell.franken.de
Running in SSH2 compatibility mode
Host key not found in 'D:\Program Files\Netscape\Users\mnbokaem\mindterm\hos
s\key_22_shell.franken.de.pub'

Connected to server running SSH-1.99-OpenSSH_2.2.0p1

shell.franken.de login: mnbokaem
mnbokaem@shell.franken.de's password: *****
Last login: Sun Nov 26 05:37:35 2000 from cr889943-b.flfrdl.on.wave.home.com

WIR HATTEN EINEN ERFOLGREICHEN EINBRUCH AUF shell.franken.de!

Ein wohl niederlaendischer Cracker hat sich ueber einen remote-root-exploit
root-Rechte auf dieser Maschine verschafft. Wir sind dabei, shell.franken.de
komplett neu zu installieren, es kann daher sein, dass einzelne Teile
des Servers nicht funktionieren

- LaForge <laforge@sunbeam.franken.de>

[shell.franken.de] [5:37] [~] >
```

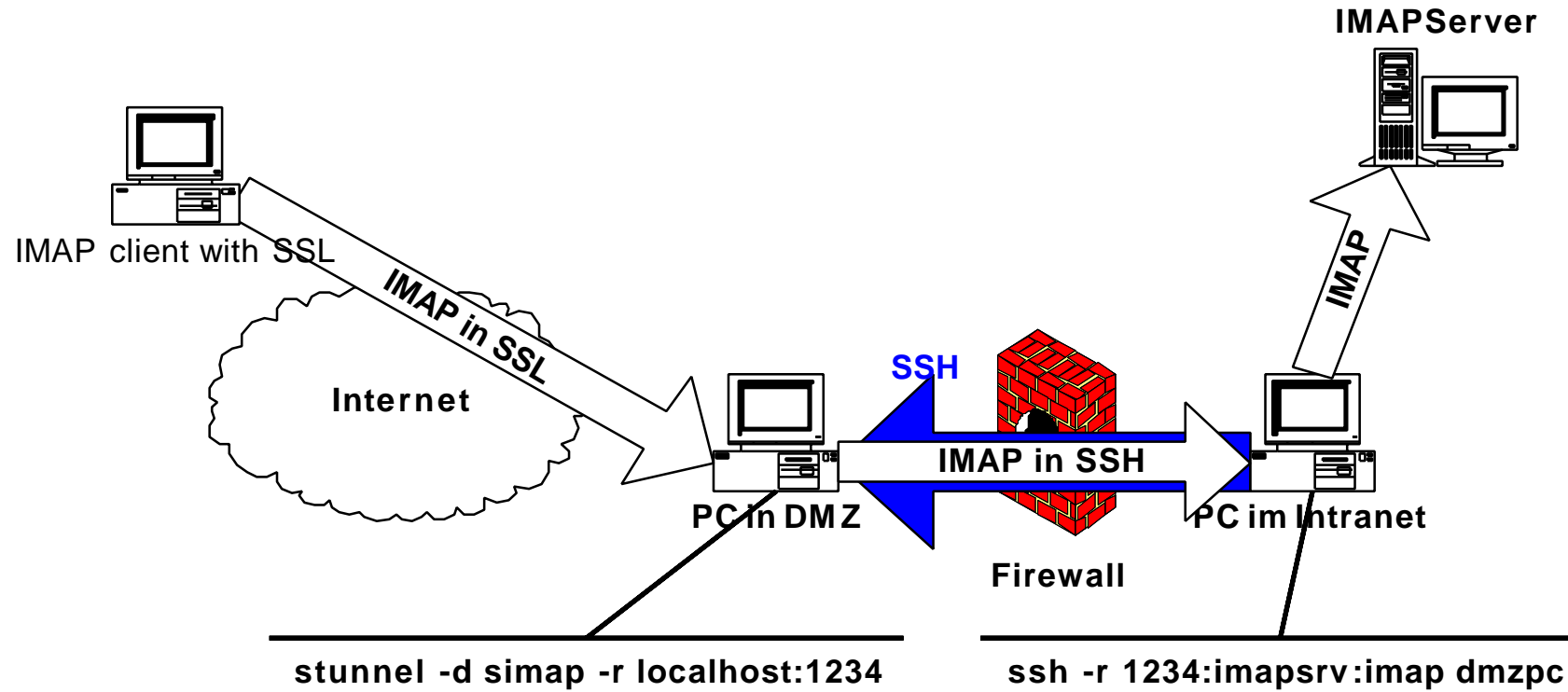
Overlaid on the terminal is a 'MindTerm - Basic Tunnels Setup' dialog box. It shows 'Current local tunnels:' with a list containing 'local: 25 -> remote: mail-n.franken.de/25'. Below this, there are input fields for 'Local port:' (25), 'Protocol:' (smtp), 'Remote host:' (mail-n.franken.de), and 'Remote port:' (25). There are 'Add' and 'Delete' buttons next to the remote port field, and a 'Close Dialog' button at the bottom. A signature bar at the bottom of the dialog reads 'Signed by: Mindbright Technology AB'.

# Zugriff auf Firmenserver



- Szenario: Zugriff von aussen auf einen Server im Intranet der selbst kein SSL beherrscht
  - SSH-tunnel von einem Rechner im Intranet zu einem Rechner im Internet (DMZ der Firma). Ein remote-port des DMZ-Rechners wird zum Intranet-Server geleitet
  - stunnel laeuft auf dem DMZ-Rechner, nimmt eine SSL-Verbindung an und leitet sie auf den weitergeleiteten Port
- Resultat: Ein Client aus dem Internet kann mit SSL auf einen Dienst im Intranet zugreifen

# Firmenserver II



# Zusammenfassung



- ‚richtige Sicherheit‘ in der Infrastruktur ist effizienter, flexibler und oft nicht verfügbar
- Mit SSL und insbesondere SSH kann man die meisten Probleme auch ‚ad-hoc‘ lösen
  
- Links:
  - Links zu freien SSH-implementationen: <http://www.freessh.org/>
  - Stunnel: <http://www.stunnel.org/>