WLAN-Security

Thomas Jampen
<jampen@cryptography.ch>

Thomas Jampen
<jampen@cryptography.ch>

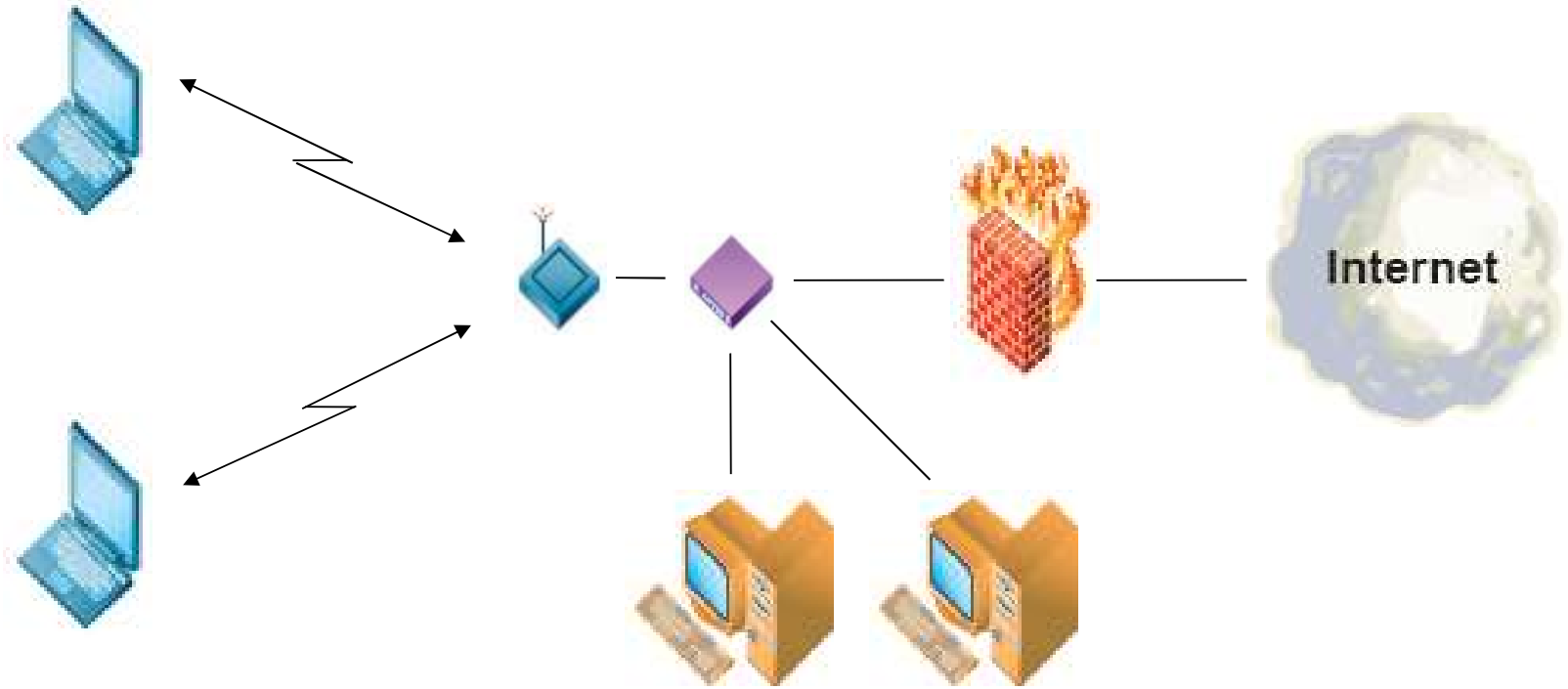
Ablauf

- Einleitung
- Ungesicherte Verbindungen
- MAC-Adressen Filter
- WEP: Standard, Schwächen, Angriffe
- WEP: Hacking Demo
- Eine sichere Alternative: IPsec

Thomas Jampen
<jampen@cryptography.ch>

Einleitung

Drahtlose Netzwerke laden zum Mithören ein...



Thomas Jampen
<jampen@cryptography.ch>

Ungesicherte Verbindungen

- Alle übertragenen Daten können mitgehört werden.
  -> Demo (Ethereal [1])
- Internet/Netzwerk-Anbindung kann verwendet und missbraucht werden.
- Verwundbare Clients können trotz Firewall/NAT direkt angegriffen werden.

Thomas Jampen
<jampen@cryptography.ch>

MAC-Adressen Filter

- Viele Access Points bieten die Möglichkeit, nur bestimmte WLAN-Karten zuzulassen (basierend auf ihrer MAC-Adresse).
- Schutz nur genügend, solange kein Client online ist (MAC unbekannt).
- MAC-Adresse kann gesnifft und softwaremässig angepasst werden!
  -> Demo

Thomas Jampen
<jampen@cryptography.ch>

Wired Equivalent Privacy

- Wired Equivalent Privacy (WEP)
- WEP ist optionaler Teil des 802.11 Standards für drahtlose Netzwerke.
- 3 Hauptziele:
  - Geheimhaltung
  - Zugriffssteuerung
  - Datenintegrität
  - (selbst synchonisierend)
  - (effizient: Hardware- oder Software-Lösung)

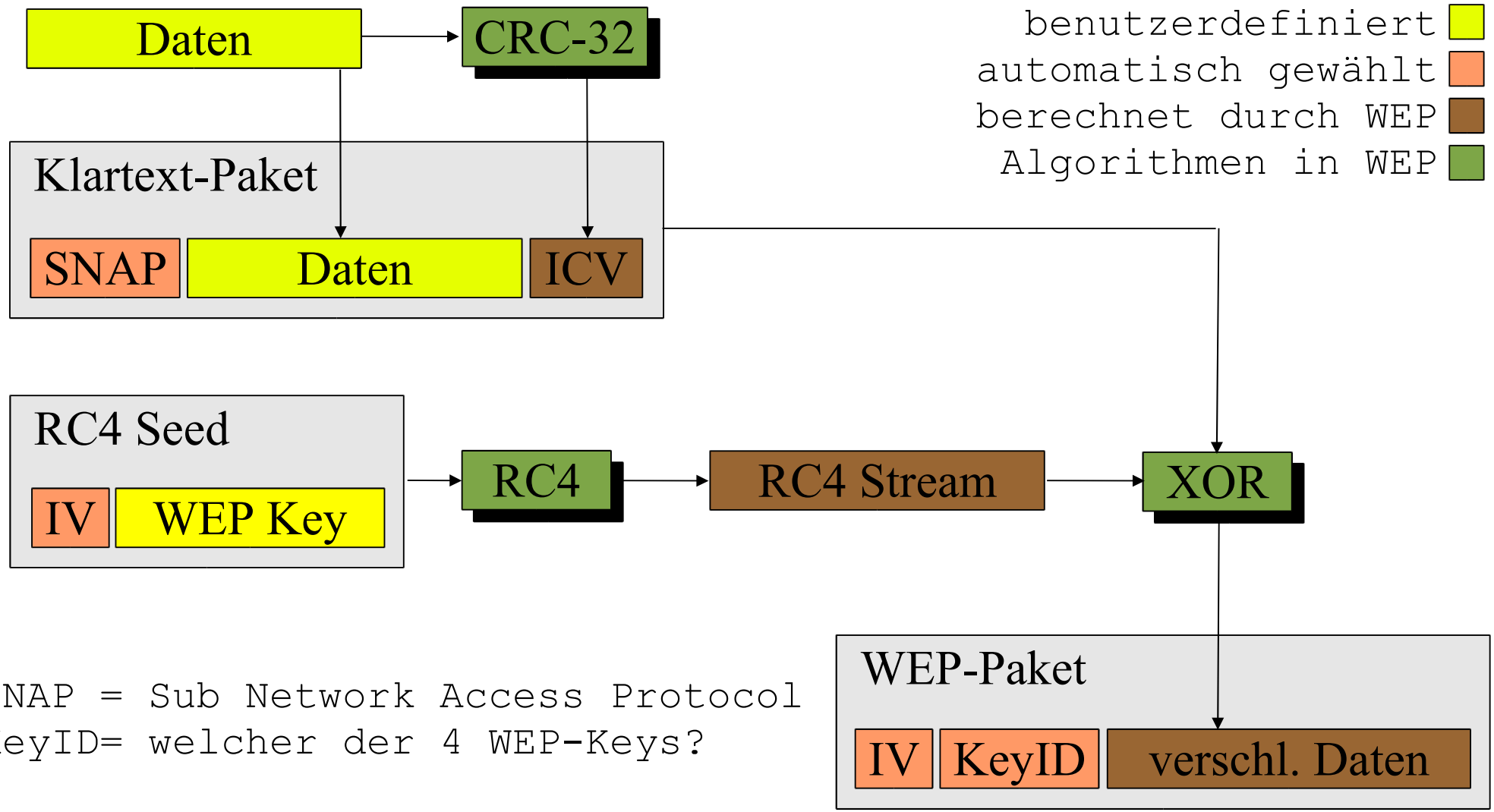
Thomas Jampen
<jampen@cryptography.ch>

WEP - Details

- Link-Layer Daten werden verschlüsselt
- Algorithmen
  - RC4 als Pseudo-Zufallszahlengenerator
  - XOR für die Verschlüsselung (Daten/Zufallszahl)
  - CRC-32 für Integrity Check Value (ICV)
- Schlüssellänge: 64 bit
  - 40 bit effektiv für WEP-Schlüssel
    -> 5 ASCII-Zeichen oder 10 Hex-Zeichen
  - 24 bit Initialisationsvektor (IV)
  - leicht erweiterbar auf 128 bit (104 bit, 24 bit)

Thomas Jampen
<jampen@cryptography.ch>

WEP - Algorithmus



SNAP = Sub Network Access Protocol
KeyID= welcher der 4 WEP-Keys?

Thomas Jampen
<jampen@cryptography.ch>

WEP - Schwächen

- Schlüsselverwaltung
  - im Standard nicht spezifiziert
  - meist 1 lang lebiger Schlüssel, fix codiert in allen Stationen (Access Points und Clients)
- Schlüssellänge
  - Standard definiert nur 40 bit (wegen US-Exportbeschränkungen)

Thomas Jampen
<jampen@cryptography.ch>

WEP – Schwächen 2

- Authentifizierung
  - „Open System“ lässt jeden Client zu, ausser die MAC-Adresse sei gesperrt
  - „Shared Key“ setzt Kenntnis des WEP-Keys voraus (Challenge/Response)
  - „Open System“ ist sicherer als „Shared Key“
    -> Attacker kennt Challenge und die verschlüsselte Response!
    -> Er kann nun jede Challenge beantworten, da er ja denselben IV wählen kann

Thomas Jampen
<jampen@cryptography.ch>

WEP – Schwächen 3

- IV ist zu klein (24 bit)
  - nur 16'777'216 verschiedene RC4 Streams (unabhängig von der Schlüssellänge!)
  - Standard definiert nicht, wie IV gewählt wird (inkrementieren oder zufällig)

    IV inkrementieren ist schlecht:
    -> 100% Kollisions-Wahrscheinlichkeit, wenn 2 APs senden!

    IV zufällig wählen ist ebenso schlecht:
    -> 50% Kollisions-Wahrscheinlichkeit, nach 4823 übertragenen Paketen

Thomas Jampen
<jampen@cryptography.ch>

WEP – Schwächen 4

- Integritätsprüf-Algorithmus unpassend
  - CRC-32 ist gut zur Aufspürung von Übertragungsfehlern, aber schlecht als kryptografische Hash-Funktion
  - CRC-32 (wie auch RC4) ist eine lineare Funktion
    -> CRC32(a) XOR CRC32(b) = CRC32(a XOR b)

    -> Angreifer kann WEP-Paket manipulieren und relativ problemlos sicherstellen, dass die Checksumme stimmt.

Thomas Jampen
<jampen@cryptography.ch>

WEP – Schwächen 5

- RC4 hat “schwache” Schlüssel
  - Zusammenhang zwischen Input und Output von RC4 manchmal grösser als erwünscht
  - ca. 9'000 der 16 Mio IV sind interessant
    -> Interessante IV sind von der Form
    (B + 3, 255, X)
    wobei B das zu berechnende Byte des WEP-Schlüssel und X zufällig ist
  - 2'000 – 4'000 interessante Pakete lassen schnell auf WEP-Schlüssel schliessen

Thomas Jampen
<jampen@cryptography.ch>

WEP - Angriffe

- Known-Plaintext-Attack
  - Angreifer sendet ein Email an Opfer.
  - Für 2 Pakete mit dem selben IV, gilt:
    XOR(Plaintext-Pakete) = XOR(Ciphertext-Pakete)
  - Der Angreifer kennt (errät) Plaintext-Paket. Wenn er das verschlüsselte Paket sniffen kann, errechnet er den RC4 Stream für diesen IV und kann somit alle mit dem selbem IV verschlüsselten Pakete entschlüsseln!

Thomas Jampen
<jampen@cryptography.ch>

WEP – Angriffe 2

- Gefälschte Pakete senden
  - Pakete können gefälscht (verändert) und mit richtiger Prüfsumme versehen werden, wenn die Klartext-Differenz bekannt ist.
- Dictionary-Attack
  - WEP-Keys sind häufig schlechte Passwörter
    -> Ein Angreifer kann gesniffte Pakete “entschlüsseln“ und sehr schnell überprüfen, ob der Key korrekt war (der SNAP Header beginnt beispielsweise immer mit 0xAA.
    -> WepAttack [2]

Thomas Jampen
<jampen@cryptography.ch>

WEP – Angriffe 3

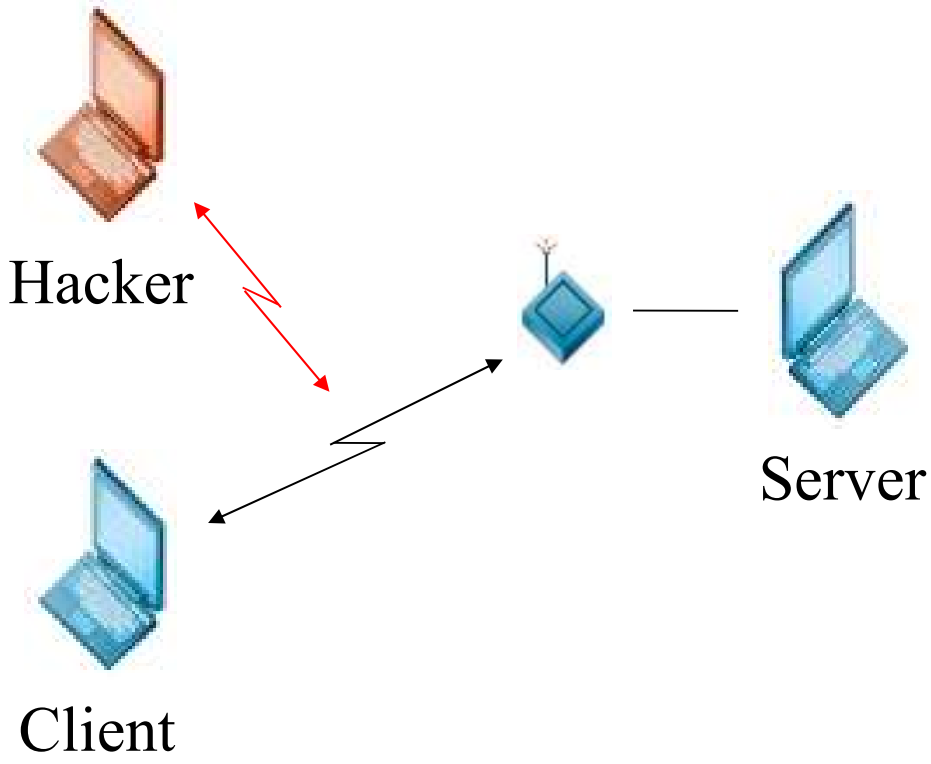
- Interessante Pakete auswerten
  - WLAN-Verkehr sniffen und interessante Pakete (schwache IVs für RC4) ausfiltern und auswerten, um schrittweise auf den WEP-Key schliessen zu können.
  - 40 bit Schlüssel in wenigen Stunden geknackt!
  - 104 bit Schlüssel in wenigen Tagen geknackt!

  -> Airsnort [3], Kismet [4]

Thomas Jampen
<jampen@cryptography.ch>

WEP – Hacking Demo

- Knoppix-CD [5] und Laptop mit unterstützter WLAN-Karte reicht!
- Demo-Szenario:



Thomas Jampen
<jampen@cryptography.ch>

Alternative: IPsec

- IP Security: RFC 2401
  Sicherheit auf Stufe IP-Layer
- 2 wichtige Protokolle
  - Authentication Header (AH) (RFC 2402)
    -> Daten-Integrität und Sender Authentizität
  - Encapsulating Security Payload (ESP) (RFC 2406)
    -> Verschlüsselung
- Basierend auf „Shared Key“
  - Internet Key Exchange (IKE) nötig (RFC 2409)
  - Asymmetrische Verfahren/Zertifikate

Thomas Jampen
<jampen@cryptography.ch>

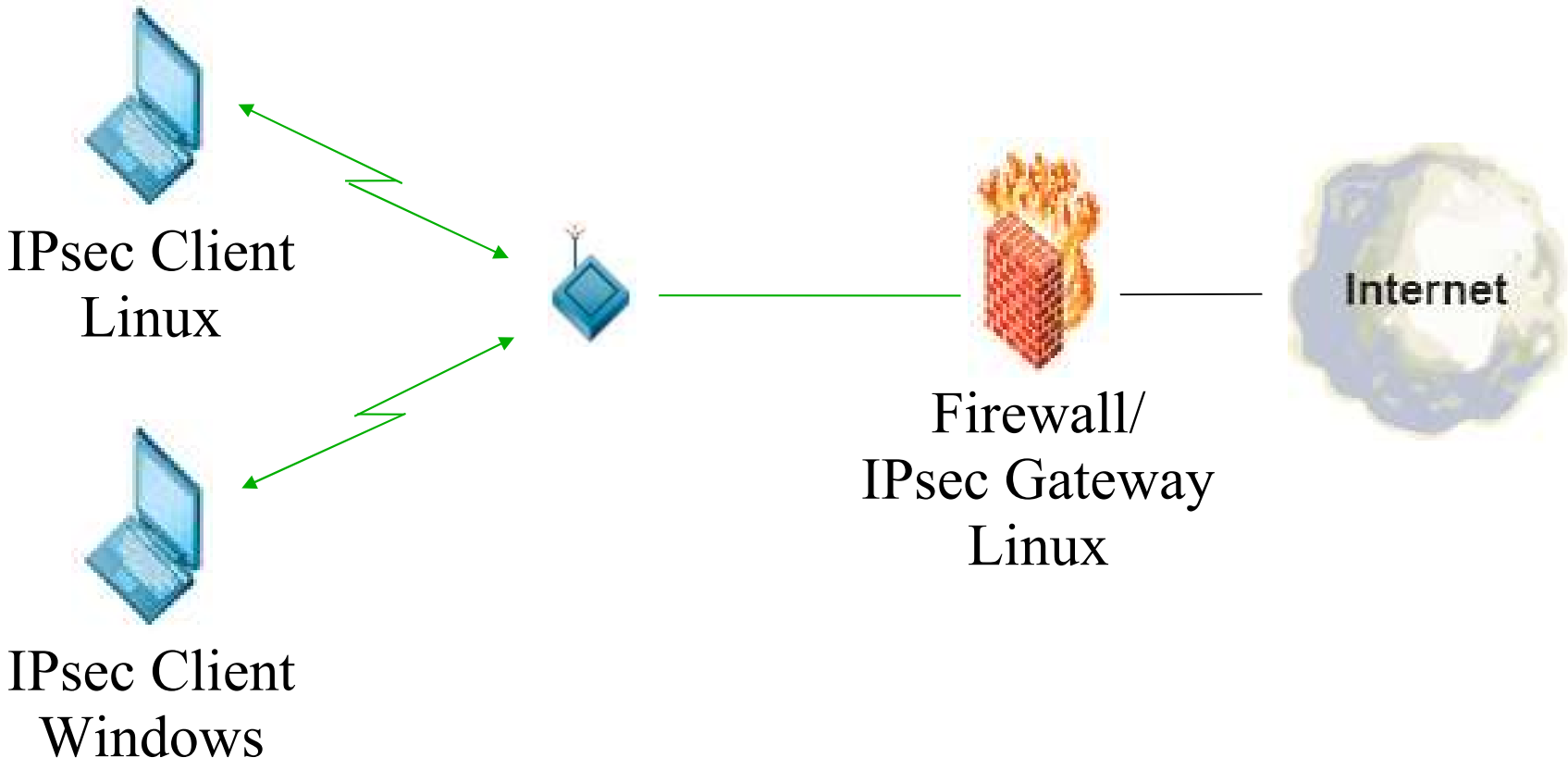
IPsec für Linux

- Kernel < 2.6
  - FreeSWAN-Implementation
  - erweiterte Versionen basierend auf FreeSWAN mit diversen Patches (StrongSWAN und OpenSWAN)
    - NAT Traversal
    - DHCP Relay
    - X509 Zertifikate
  - FreeSWAN-Entwicklung eingestellt seit Kernel 2.6
- Kernel >= 2.6
  - KLIPS (Kernel IPsec für ESP, AH, Paket-Handling)
  - Pluto (IKE Daemon) von OpenSWAN oder StrongSWAN

Thomas Jampen
<jampen@cryptography.ch>

IPsec - Home WLAN Szenario

Einsatz im drahtlosen Heimnetzwerk



Thomas Jampen
<jampen@cryptography.ch>

Linux IPsec Installation

- Kernel 2.6 mit folgenden Modulen
  - af_key, ah4, esp4, ipcomp, xfrm_user
  - Module für Verschlüsselung und Hash Alg.
- StrongSWAN (nur Pluto) (Details [6])
  - Makefile anpassen (falls gewünscht)
  - make programs
  - make install
- Zertifikate erstellen (IPsec-Howto [7])

Thomas Jampen
<jampen@cryptography.ch>

Gateway IPsec Konfiguration

```
config setup
   interfaces=”ipsec0=wlan0”   # wlan0 symbolisch
                               # für Device mit
conn %default                  # Anschluss an AP
   keyingtries=1
   compress=yes
   disablearrivalcheck=no
   authby=rsasig
   leftrsasigkey=%cert
   rightrsasigkey=%cert
```

Thomas Jampen
<jampen@cryptography.ch>

Gateway IPsec Konfiguration 2

```
conn roadwarrior-net
   leftsubnet=0.0.0.0/0
   also=roadwarrior

conn roadwarrior
   left=192.168.1.1          # Gateway IP
   leftcert=gateway.pem
   right=%any                # jeder mit gültigem,
   auto=add                  # von CA ausgestelltem
   pfs=yes                   # Zertifikat darf
```

Thomas Jampen
<jampen@cryptography.ch>

Client IPsec Konfiguration

```
config setup
   interfaces=%defaultroute

conn %default
   keyingtries=1
   compress=yes
   authby=rsasig
   leftrsasigkey=%cert
   rightrsasigkey=%cert
```

Thomas Jampen
<jampen@cryptography.ch>

Client IPsec Konfiguration 2

```
conn roadwarrior-net
   rightsubnet=0.0.0.0/0
   also=roadwarrior

conn roadwarrior
   right=192.168.1.1
   rightcert=gateway.pem
   rightid="C=CH, ST=Bern, L=Bern, O=LugBE, CN=GW"
   left=%defaultroute
   leftcert=linuxclient.pem
   auto=add
```

Thomas Jampen
<jampen@cryptography.ch>

Win2k/XP IPsec Installation

- Win2k/XP haben IPsec Stack
- Support Tools installieren
  -> Achtung WinXP SP2 [8]
- Zertifikate im *.p12 exportieren (Linux) und unter Windows importieren
- IPsec-Tool entpacken (DL + Infos [9])
  -> konfiguriert Registry gemäss ipsec.conf (Syntax wie bei Linux)
- WinXP Firewall: UDP Port 500 erlauben

Thomas Jampen
<jampen@cryptography.ch>

Windows IPsec Konfiguration

```
conn roadwarrior
    left=%any
    mac=11-22-33-44-55-66
    right=192.168.1.1
    rightca="C=CH,S=Bern,L=Bern,O=LugBE,CN=CA"
    network=lan
    auto=start
    pfs=yes
```

Achtung: ST=Bern -> S=Bern (siehe auch [7] und [9])

mac ev. nötig, wenn mehrere Netzwerkkarten vorhanden

Thomas Jampen
<jampen@cryptography.ch>

Windows IPsec Konfiguration 2

```
conn roadwarrior-net
    left=%any
    mac=11-22-33-44-55-66
    right=192.168.1.1
    rightsubnet=*
    rightca="C=CH,S=Bern,L=Bern,O=LugBE,CN=CA"
    network=lan
    auto=start
    pfs=yes
```

Thomas Jampen
<jampen@cryptography.ch>

IPsec Tunnel starten

- Linux-Client

  ```
  linux:~# ipsec auto --up roadwarrior
  linux:~# ipsec auto --up roadwarrior-net
  ```

- Windows-Client

  C:\Programme\IPsec\ipsec.exe ausführen (beispielsweise im Autostart)

  Die ersten 1-4 (Ping-)Pakete lösen „Negotiating IP-Security“ aus, dann steht die Verbindung.

Thomas Jampen
<jampen@cryptography.ch>

Linux IPsec-Firewallregeln

- Was muss zugelassen werden?
  - Protokoll 50 (ESP) und 51 (AH) zulassen (das sind keine Ports!)
  - UDP-Port 500 zulassen
- Kernel 2.4 mit FreeSWAN erzeugt virtuelle ipsecX Devices
  - z.B. entspricht eth0 dem virtuellen ipsec0
  - Es kann jeglicher Verkehr über eth0 verboten, aber Pakete über ipsec0 zugelassen werden, somit haben nur Clients mit gültigem Zertifikat Zugriff zum Netzwerk/Internet.

Thomas Jampen
<jampen@cryptography.ch>

Linux IPsec-Firewallregeln 2

- Kernel 2.6 ohne virtuelle ipsecX
  - eingehendes ESP-Paket durchläuft die Hooks

PRE_ROUTING	(verschlüsselt)
LOCAL_IN	(verschlüsselt)
PRE_ROUTING	(Klartext)
LOCAL_IN/FORWARD	(Klartext)

  - abgehendes ESP-Paket

LOCAL_OUT/FORWARD	(Klartext)
POST_ROUTING	(Klartext)
LOCAL_OUT	(verschlüsselt)
POST_ROUTING	(verschlüsselt)

Thomas Jampen
<jampen@cryptography.ch>

Linux IPsec-Firewallregeln 3

- Wie stellt man fest, ob ein eingehendes TCP/UDP-Paket vorher verschlüsselt war?
- Wie stellt man sicher, dass nur verschlüsselte Pakete versendet werden können?
- Iptables hat eine „mangle“-Table, Pakete können markiert und später auf die Marke hin überprüft werden.

Thomas Jampen
<jampen@cryptography.ch>

Linux IPsec-Firewallregeln 4

Marke setzen:

```
iptables -t mangle -A PREROUTING -i $WLAN_DEV -p esp      \
         -j MARK --set-mark 1
```

Marke prüfen:

```
iptables -A FORWARD -i $WLAN_DEV -o $EXT_DEV -s $WLAN_NET \
         -m state --state NEW -m mark --mark 1 -j ACCEPT
```

Alle anderen neuen Verbindungen verbieten, etablierte und verwandte können zugelassen werden.

Analog für eingehende (INPUT) und für ausgehende (OUTPUT).

Thomas Jampen
<jampen@cryptography.ch>

Links

[1] http://www.ethereal.com/

[2] http://sourceforge.net/projects/wepattack

http://sourceforge.net/projects/wepdecrypt

[3] http://airsnort.shmoo.com/

[4] http://www.kismetwireless.net/

[5] http://www.knoppix.org/

[6] http://www.stronswan.org/

[7] http://www.natecarlson.com/linux/ipsec-x509.php

[8] http://support.microsoft.com/default.aspx?scid=kb;en-us;838079

[9] http://vpn.ebootis.de/