FreeS/WAN

Thomas Bader
thomasb@trash.net

Überblick

- Einführung
- Authentifizierung
- Bestandteile von FreeS/WAN
- Installation
- Zertifikate generieren
- Konfigurationsgrundlagen

Einführung

- Freie Implementierung von IPSEC
- Für Linux
- GPL
- Auf allen IP Netzen nutzbar
- Keine Anpassung im Userspace nötig

Performance

- Mittelklasse-Rechner:
  - Linux 2.2.20 mit OpenWall
  - Pentium MMX 233Mhz
  - 128MByte RAM
  - Adaptec 2940, SCSI II Disks
  - 3x Realtek RTL8139
  - Gleichzeitig Gateway für 40 Client Rechner
- Getestet mit:
  - Zwei Wireless Clients (je 11Mbit) gleichzeitig
  - Mit rsync auf Rechner an anderem Interface
- Fazit: Ohne Probleme möglich
  - CPU Auslastung unter 30%

Authentifizierung

Shared Secret

- Gemeinsames Kennwort auf beiden Seiten
- "Einfach" in der Handhabung
- Bei Kompromittierung müssen u.U. alle Teilnehmer mit neuem Kennwort ausgestattet werden
- Kennwörter machen Brute Force relativ einfach

Zertifikate

- Jeder Teilnehmer hat ein eigenes Zertifikat
- Zertifikate können selektiv zurückgerufen werden
- Brute Force Attacken werden erschwert

Bestandteile einer Zertifikatverwaltung

- Certification Authority (CA)
- Zertifikat der CA (2048 Bit)
- Für jeden Rechner ein Zertifikat (1024 Bit)
- Certificate Revocation List (CRL)
- OpenSSL (oder anderes Toolkit)

Bestandteile von FreeS/WAN

KLIPS

- Kernel IPSEC Support

ipsec

- IPSEC Utilities aufrufen
- Manual Page: ipsec(8)

pluto

- IKE (IPSEC Key Exchange) Daemon
- Kann über *ipsec whack* gesteuert werden
- Zuständig für automatischen Schlüsselaustausch
- Manual Page: ipsec_pluto(8)

klipsdebug

- KLIPS Debugging Features und Level setzen
- *ipsec klipsdebug*
- Manual Page: ipsec_klipsdebug(8)

barf

- Debugging Informationen anzeigen
- *ipsec barf*
- Manual Page: ipsec_barf(8)

Installation

Voraussetzungen

- Diverse Tools
  - gcc oder egcs
  - Assembler und Linker (zB. bin86)
  - make und patch
- Object- und Header-Files aller Libraries
  - glibc et. al.
  - GMP (GNU Multi Precision) Library
- Kernel Sourcen
  - Konfiguriert, kompiliert und getestet

Quellcode downloaden

```
# cd /usr/src
# wget ftp://ftp.xs4all.nl/pub/crypto/freeswan/LATEST.tar.gz
# wget http://www.strongsec.com/freeswan/ \
      x509patch-0.9.12-freeswan-1.97.tar.gz
# tar xfz LATEST.tar.gz
# tar xfz x509patch-0.9.12-freeswan-1.97.tar.gz
```

FreeS/WAN patchen

```
# cd /usr/src/freeswan-1.97
# patch -p1 < ../x509patch-0.9.12-freeswan-1.97/freeswan.diff
```

Kernel patchen

- Patch einspielen

```
# cd /usr/src/freeswan-1.97
# make menugo          # 'make menuconfig'
# make xgo             # 'make xconfig'
# make ogo             # 'make config'
# make oldgo           # 'make oldconfig'
```

- Kernel konfigurieren
  - Unter "Networking options" sind FreeS/WAN Optionen
  - FreeS/WAN fest in den Kernel
  - "Advanced Router" abstellen
  - Speichern
- Danach wird der Kernel kompiliert

Nach dem Kompilieren

- Es wurden einige Dinge installiert:
  - User-Level Utilities sind in *<prefix>/lib/ipsec*
  - *ipsec* Kommando in *<prefix>/sbin/ipsec*
  - Manual Pages in *<prefix>/man/man[1-8]*
  - Init Skript */etc/init.d/ipsec*
- Für die Konfiguration ist zuständig:
  - */etc/ipsec.conf*
  - */etc/ipsec.secrets*
  - */etc/ipsec.d/*
- Jetzt muss der Kernel neu installiert und gebootet werden

Nach dem Reboot

- Wir können überprüfen, ob FreeS/WAN läuft

```
# ifconfig ipsec0
ipsec0    Link encap:Ethernet  HWaddr 00:30:4F:0A:E8:08
          inet addr:195.144.58.34  Mask:255.255.255.248
[....]
# ps axf
  547 ?  S 0:00 sh /usr/local/lib/ipsec/_plutorun
  551 ?  S 0:00  \_ sh /usr/local/lib/ipsec/_plutorun
  554 ?  S 0:00  |   \_ /usr/local/lib/ipsec/pluto
  552 ?  S 0:00  \_ sh /usr/local/lib/ipsec/_plutoload
  548 ?  S 0:00 logger -p daemon.error -t ipsec__plutorun
# ipsec --version
Linux FreeS/WAN 1.97
See `ipsec --copyright' for copyright information.
```

Zertifikate generieren

CA Zertifikat generieren

```
# mkdir /root/crypto
# chmod 700 /root/crypto
# cd /root/crypto
# openssl req -x509 -days 1460 -newkey rsa:2048 \
          -keyout caKey.pem -out caCert.pem
[Es werden die Eigenschaften und ein Mantra gefragt]
```

- Erstellt:
  - 2048 Bit RSA Private Key *caKey.pem*
  - Self-Signed CA Zertifikat *caCert.pem*
- Haben Gültigkeit von 4 Jahren

Eigenschaften des CA Zertifikats

```
# openssl x509 -in caCert.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=CH, ST=Zuerich, L=Zuerich, O=Fiktive CA,
           OU=The Orange Clockwork,
           CN=Vorname Nachname/Email=ca@domain.tld
        Validity
            Not Before: Jun  2 12:27:09 2002 GMT
            Not After : Jun  1 12:27:09 2006 GMT
[...]
```

CA Zertifikat installieren

```
# openssl x509 -in caCert.pem -outform DER \
          -out /etc/ipsec.d/cacerts/cacert.der
```

CA Verzeichnis erstellen

```
# mkdir -p demoCA/newcerts
# echo "00" > demoCA/serial
# touch demoCA/index.txt
```

Host Zertifikat erstellen

```
# openssl req -newkey rsa:1024 -keyout hostKey.pem \
        -out hostReq.pem
# openssl ca -in hostReq.pem -days 730 -out hostCert.pem \
        -notext -cert caCert.pem -keyfile caKey.pem
# openssl pkcs12 -export -inkey hostKey.pem \
        -in hostCert.pem -name "Name" \
        -certfile caCert.pem -caname "Fiktive CA" \
        -out hostCert.p12
```

- Lediglich *hostReq.pem* muss an die CA geschickt werden
- CA retourniert *hostCert.pem*
- Letztes Kommando generiert eine PKCS12 Datei für Windows Clients

Certificate Revokation List erstellen

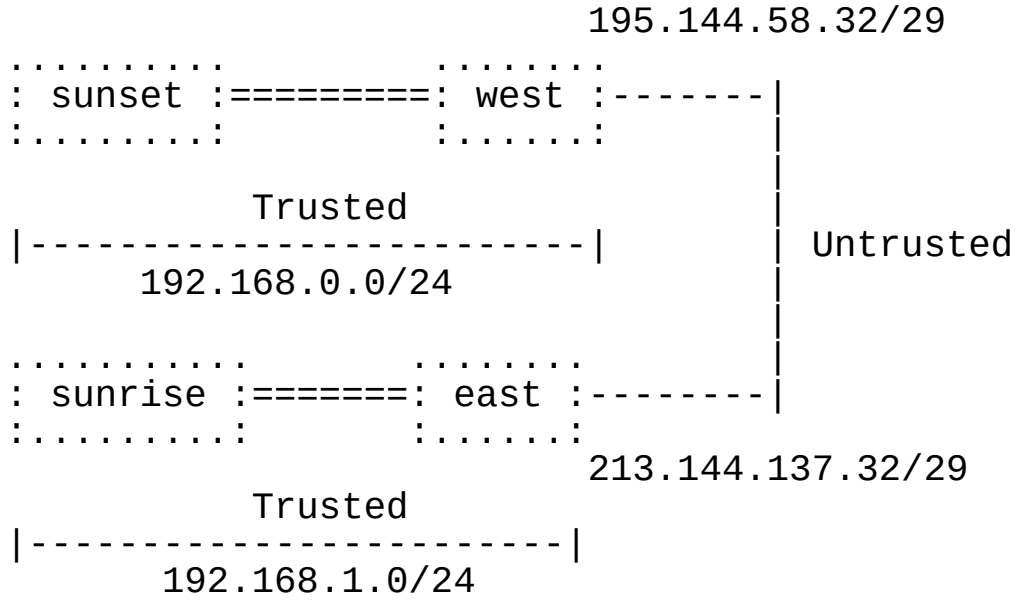
```
# openssl ca -gencrl -crldays 15 -out crl.pem \
         -cert caCert.pem -keyfile caKey.pem
# openssl crl -in crl.pem -outform DER \
         -out /etc/ipsec.d/crls/cert.crl
```

Zertifikat revoken

```
# less demoCA/index.txt
# openssl ca -revoke demoCA/newcerts/00.pem \
         -keyfile caKey.pem -cert caCert.pem
# openssl ca -gencrl -crldays 60 -out crl.pem \
         -keyfile caKey.pem -cert caCert.pem
# openssl crl -in crl.pem -noout -text
# openssl crl -in crl.pem -outform DER \
         -out /etc/ipsec.d/crls/cert.crl
```

Konfigurationsgrundlagen

Beispielnetz



Grundsätzliche Vorkehrungen

- Netzwerk prüfen **ohne** IPSEC
- Packet Forwarding enablen
  - echo "1" > /proc/sys/net/ipv4/ip_forward
- Folgender Traffic darf nicht gefiltert sein:
  - UDP Port 500 (für IKE)
  - IP Protokoll 50 (für ESP)
  - IP Protokoll 51 (für AH)

Links? Rechts?

- Beim Konfigurieren von FreeS/WAN wird immer eine Seite als linke und eine als rechte bezeichnet
- Welche Seite links oder rechts, kann selber bestimmen werden

Privaten Schlüssel installieren

```
# cat /etc/ipsec.secrets
:RSA /etc/ipsec.d/private/hostKey.pem "passphrase"
```

- Falls der Schlüssel noch transportiert werden muss:

```
# openssl genrsa -des3 -out hostKey.pem 1024
[ transportieren ]
# openssl rsa -in hostKey.pem -out hostKey.pem
```

ipsec.conf - config setup

- Auf beiden Seiten gleich

```
config setup
      interfaces=%defaultroute
      klipsdebug=none
      plutodebug=none
      plutoload=%search
      plutostart=%search
      uniqueids=yes
```

ipsec.conf - conn %default

- Für west:

```
conn %default
      # mit RSA Zertifikaten authentifizieren
      authby=rsasig
      rightrsasigkey=%cert
      # meine Seite ist links
      left=195.144.58.34
      # Wird in /etc/ipsec.d gesucht
      leftcert=mycerts/hostCert.pem
      # Verbindungen automatisch laden
      auto=add
```

- Für east:

```
conn %default
      authby=rsasig
      leftrsasigkey=%cert
      right=213.144.137.34
      rightcert=mycerts/hostCert.pem
      auto=add
```

ipsec.conf - Verbindung definieren

- Auf beiden Seiten gleich

```
conn west-east
     left=195.144.58.34
     leftnexthop=195.144.58.33
     leftsubnet=192.168.0.0/24
     right=213.144.137.34
     rightnexhop=213.144.137.33
     rightsubnet=192.168.1.0/24
     auto=start
```

Konfiguration ausprobieren

```
west:~# ipsec setup --restart
ipsec_setup: Stopping FreeS/WAN IPsec...
ipsec_setup: Starting FreeS/WAN IPsec 1.97...
west:~# ifconfig eth0 | grep 'inet addr'
          inet addr:195.144.58.34  Mask:255.255.255.248
west:~# ifconfig ipsec0 | grep 'inet addr'
          inet addr:195.144.58.34  Mask:255.255.255.248
west:~# netstat -rn
0.0.0.0         195.144.58.33 0.0.0.0         UG  0 0 0 eth0
192.168.0.0     0.0.0.0       255.255.255.0   U   0 0 0 eth1
195.144.58.32   0.0.0.0       255.255.255.248 U   0 0 0 eth0
195.144.58.32   0.0.0.0       255.255.255.248 U   0 0 0 ipsec0
192.168.1.0     195.144.58.33 255.255.255.0   U   0 0 0 ipsec0
```

Falls es nicht funktioniert...

```
$ lynx http://www.freeswan.org/freeswan_trees/ \
       freeswan-1.95/doc/trouble.html
```

Links

- http://www.freeswan.org/
- http://www.strongsec.com/freeswan/