Praktische Anleitung zu Konfiguration von IPSEC Verbindungen

mittels FreeS/WAN und PGPnet

Torsten Höfler

htor@informatik.tu-chemnitz.de
hoefler@delta.de

Inhaltsverzeichnis

1 Konfiguration von IPSEC mittels des FreeS/WAN

1.1 Benötigte Software

- FreeS/WAN Software www.freeswan.org mit x509 und _confread Patch
  hier wurde Version 1.91 verwendet
- Kernelsourcen www.kernel.org
  hier wurde Version 2.4.9 verwendet
- Keyextractor keyextractor binary
  (nur bei Anbindung von Windows Clients mit PGPnet)

1.2 Installation der FreeS/WAN Software

- Konfiguration des Kernel Source-trees (siehe Kernel-HOWTO)

```
apollo:/usr/src/linux # make menuconfig
```

- Entpacken des tar-archives in ein beliebiges source-Verzeichnis

```
apollo:/usr/src # tar xzf freeswan-1.91.tar.gz
```

- Hinzufügen aller benötigten Patches x509-Patch

```
apollo:/usr/src/freeswan-1.91 # patch -p0 < \
 ../x509patch-0.9.1-freeswan-1.91/pluto.diff

apollo:/usr/src/freeswan-1.91/utils # patch -p0 < ../../\
 x509patch-0.9.1-freeswan-1.91/fswcert/_confread.patch
```

- Installation nach der INSTALL-Anleitung im freeswan-Verzeichnis (/usr/src/freeswan-1.91)

```
apollo:/usr/src/freeswan-1.91 # make menugo
apollo:/usr/src/freeswan-1.91 # make install
```

- Installation der (neuen) Kernel Module

```
apollo:/usr/src/linux # make modules_install
```

1.3 Konfigurieren des FreeS/WAN Servers

Mit der hier angeführen Konfiguration wird ein typisches „Road Warrior“ Setup beschrieben, wobei sich jeder Client, der sich authentifizirt am Server anmelden kann. Um eine statische Konfiguration aufzusetzen muss bei right anstelle von %any (oder 0.0.0.0) eine feste IP eingetragen werden. In allen Konfigurationsdateien wird der Server immer als „left“ bezeichnet.

1.3.1 Erstellen eines neuen RSA Schlüsselpaares

Zuerst sollte für den Server ein neues RSA Schlüsselpaar generiert werden. Dies kann z.B. mit

```
apollo:/usr/src/freeswan-1.91 # ipsec rsasigkey 1024
```

mit einer Schlüssellänge von 1024 Bits erzeugt werden. Der erzeugte Output muss in die Datei /etc/ipsec.secrets an die Stelle : RSA { ... [output] ... } eingefügt werden (ansonsten sollte man die Datei ipsec.secrets nicht weiter verändern). Zu Beachten ist hier, dass die schliessende eckige Klammer unbedingt eingerückt sein muss!
Der erste Schlüssel stellt den Public-Key dar, der später beim nächsten Punkt wichtig wird.
Beispiel einer ipsec.secrets:

```
: RSA {
        Modulus:0x...
        PublicExponent: 0x...
        PrivateExponent: 0x...
        Prime1: 0x...
        Prime2: 0x...
        Exponent1: 0x...
        Exponent2: 0x...
        Coefficient: 0x...
        }
```

1.3.2 Die Konfigurationsdatei /etc/ipsec.conf (Server)

/etc/ipsec.conf ist die zentrale (und einzige) Konfigurationsmöglichkeit des FreeS/WAN IPSEC Paketes. Sie teilt sich in zwei Hauptbereiche, einmal einen config Bereich (für generelle Konfigurationseinstellungen) und einen conn Bereich (für Einstellungen, die Verbindungen betreffen). Beispiel einer ipsec.conf (Server):

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

conn %default
        keyingtries=0
        authby=rsasig

conn sample-conn
        left=xxx.xxx.xxx.xxx
        leftid=...
        leftrsasigkey=0x...
        right=%any
        rightid=...
        rightrsasigkey=0x...
        auto=add
```

Zu beachten ist hierbei, dass die Einrückung (z.B. mittels Tabulator) zur Sektionsunterteilung eine wichtige Rolle spielt.

Beschreibung der einzelnen Parameter:

interfaces Definition der Netzwerkinterfaces, an die sich die virtuellen ipsecxx Devices binden %defaultroute ist Standard, man kann die devices mittels "'ipsec0=eth0 ..." fest an bestimmte Interfaces knüpfen.

klipsdebug Debug-Einstellungen von Klips (Kernel-Modul) entweder „none" für (fast) keine, oder „all" fuer alles

plutodebug Debug-Einstellungen von Pluto entweder none für keine, oder all fuer alles

plutoload Welche connections sollen automatisch geladen werden

plutostart Welche connections sollen automatisch gestartet werden

keyingretries Anzahl der Verbindugsversuche bei Misserfolg (0 bedeutet unendlich)

authby Authentifizierungsverfahren (z.B. rsasig=RSA Key)

left Netzwerkadresse des Servers (ip)

leftsubnet Subnetzt auf der linken Seite (Routing betreffend)

leftid Bezeichner für Server (falls ein „@" vornagestellt ist, wird die angegebene Adresse nicht aufgelöst, diese Einstellung wird empfohlen)

leftrsasigkey Öffentlicher RSA Schlüssel des Servers (0x...) siehe Punkt 1.3.1

right... siehe Einstellungen für left, diese betreffen jedoch den Client

auto Anweisung für Puto, ob die Verbindung beim starten geladen werden soll (add), oder sofort gestartet wird (start)

1.4 Konfiguration eines FreeS/WAN Clients (Linux)

Das Erstellen eines RSA Schlüsselpaares für den Client läuft analog zu Punkt 1.3.1. Nach dem Erstellen des Schlüssels sollte der öffentliche Teil (pubkey) in der ipsec.conf des Servers und des Clients als leftkey eingestellt werden.

1.4.1 Die Konfigurationsdatei /etc/ipsec.conf (Client)

Analog zu Punkt .
Beispiel einer ipsec.conf (Client)

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

conn %default
        keyingtries=1
        authby=rsasig
```

```
conn sample-conn
        left=xxx.xxx.xxx.xxx
        leftsubnet=xxx.xxx.xxx.xxx/xx
        leftid=...
        leftrsasigkey=0x...
        right=%defaultroute
        rightsubnet=xxx.xxx.xxx.xxx/xx
        rightid=...
        rightrsasigkey=0x...
        auto=add
```

Beschreibung siehe Punkt 1.3.2.

1.4.2 Starten des Tunnels

Nun muss man auf Server und Client des ipsec-Paket starten.

```
z.B. mit apollo:~ # /etc/init.d/ipsec start
```

Nun kann man den Tunnel vom Client aus mittels

```
apollo:~ # ipsec auto --up sample-conn
```

eine Verbindung aufbauen.

1.4.3 Testen der Verbindung (Verschlüsselung)

Ob auch wirklich eine verschlüsselte Verbindung verwendet wird kann man am Besten mittels tcpdump prüfen. Egal, welche ip-Pakete man durch den Tunnel schickt, tcpdump sollte nur ip-proto-50 anzeigen. (Am Ethernetdevice werden die verschlüsselten/eingepackten Pakete angezeigt, am ipsecx-Device koennen die selben Pakete unverschlüsselt mitgesnifft werden).

1.5 Konfiguration eines PGPnet Clients (Windows)

Für die Anbindung von Windows Clients an FreeS/WAN ist PGPnet in der Version 6.5.8 in Verbindung mit dem Keyextractor (siehe Punkt 1.1) gut geeignet.

1.5.1 Erzeugen der Schlüssel

Zuerst müssen zwei Schlüssel (für FreeS/WAN und für PGPnet) erzeugt werden. Dies muss für jeden Schlüssel einzeln geschehen. Ich empfehle folgendes Vorgehen: (eine genauere Anleitung mit Screenshots wird es noch in einem extra Dokument geben) Das Erzeugen der Keys ist derzeit leider nur mit PGPnet Version 6.5.8 möglich, man kann jedoch die damit erzeugten Keys auch mit aktuelleren PGPnet Versionen (z.B. 7.0.3) verwenden (siehe Screenshots-Dokument).

- Auf dem Windows Rechner eine Leere Diskette mit zwei Verzeichnissen (freeswan + pgpnet) erzeugen
- in PGPnet alle (wichtig!) vorhandenen Schlüssel löschen (kann ja man vorher sichern)
- mit PGPnet einen neuen Schlüssel (RSA, 1024 Bit) erzeugen, den Fingerprint (Hexadezimal - unter „Key Properties“) mit auf Diskette sichern und PGPnet schliessen (Dieser Schlüssel wird später als Key für den FreeS/WAN Server verwendet)

- die Dateien pubring.pkr und secring.skr aus dem PGP Keyrings - Verzeichnis auf Diskette in den Ordner FreeS/WAN kopieren
- PGPnet starten, den Schlüssel löschen, einen neuen RSA-Key 1024-Bit erzeugen, den zweiten Fingerprint ebenfalls sichern und PGPnet wieder beenden (Dieser Key wird später als Key für PGPnet verwendet)
- die Datei pubring.pkr auf Diskette in das Verzeichnis pgpnet kopieren (falls eine andere PGPnet Version, oder PGPnet auf einem anderen Rechner verwendet wird sollte auch die secring.skr mit auf Diskette kopiert werden, um sie später mit der anderen PGPnet version wieder importieren zu können)
- die Diskette an der Linux Maschiene (z.B. unter /floppy) mounten
- die pubring.pkr des Servers (freeswan Verzeichnis) nach /etc/pgpcert.pgp kopieren (sie darf nur einen Schlüssel enthalten!)
- für beider Verzeichnisse mittels keyextractor die FreeS/WAN kompatiblem Keys erzeugen z.B.

```
apollo:/floppy/freeswan # keyextractor freeswan \
freeswan.out pubring.pkr secring.skr
```

  Achtung: in den Keyring Dateien (*pkr, *skr) darf jeweils nur ein Schlüssel enthalten sein!

- der Key sollte jetzt die von FreeS/WAN gewohnte Gestalt haben z.B.

```
apollo:/floppy/freeswan # cat freeswan.out
pubkey: #0x...
Modulus: 0x...
PublicExponent: 0x...
PrivateExponent: 0x...
Prime1: 0x...
Prime2: 0x...
Exponent1: 0x...
Exponent2: 0x...
Coefficient: 0x...
```

- nun kann das FreeS/WAN Schlüsselpaar in die Datei /etc/ipsec.secrets ähnlich zu Punkt 1.3.1 eingefügt werden , jedoch muss
  - die erste Zeile (pubkey: 0x..) auskommentiert werden
  - vor „: RSA“ muss der freeswan Fingerprint des Servers stehen

  z.B.

```
@#4094B322E44EBBDB699C4B4AD4A07808: RSA {
#pubkey: #0x...
Modulus: 0x...
PublicExponent: 0x...
PrivateExponent: 0x...
Prime1: 0x...
```

```
Prime2: 0x...
Exponent1: 0x...
Exponent2: 0x...
Coefficient: 0x...
       }
```

- in der ipsec.conf auf dem Server müssen noch beide public-keys eingetragen werden (stehen bei pubkey) bei leftrsasigkey, die des Servers (Verzeichnis freeswan) und bei rightrsasigkey die des PGPnet Clients (Verzeichnis pgpnet).
- zusätzlich muss der Hexadezimale Fingerprint der Keys (freeswan-Key bei leftid, und pgpnet-Key bei rightid in der ipsec.conf eingetragen werden z.B.

```
... [ipsec.conf] ...
leftid=@#4094B322E44EBBDB699C4B4AD4A07808
...
rightid=@#99806BEFBB553B6F55D6016EFA45014A
```

- den FreeS/WAN Server neu starten
- im PGPnet muss das Schlüsselpaar, das wir für PGPnet erstellt haben komplett (private und public) enthalten sein, während vom FreeS/WAN Schlüssel nur der öffentliche Teil gespeichert sein sollte (man kann die Keys von der Diskette erneut importieren)
- das PGPnet Schlüsselpaar muss als Vertrauenswürdig markiert werden („Implicit Trust“ bei Key Properties), und der freeswan public key muss unterschrieben werden, und anschliessend auch auf trusted gesetzt werden.

1.5.2 Konfigurieren von PGPnet

1. PGPnet Version 6.5.8
   - Importieren aller Schlüssel (pgpnet public und private, freeswan nur public)
   - Unterschreiben der Schlüssel (und diese auf Trusted setzen)
   - View - Options:
     - General
       - Expert Mode
       - Allow communication with unconfigured hosts
       - Require valid authentication key
       - Setup Keys (IKE) Duration: 1h
       - Primary Keys (IPSEC) Duration: 1h
     - Authetication
       - Bei PGP Authetication den erzeugten pgpnet Key wählen
     - Advanced
       - Cipers: TripleDES
       - Hashes: SHA-1, MD5
       - Diffie-Hellmann: 1024 Bits
       - Compression: keine Auswahl
       - Proposals

- IKE:
  (a) RSA-Signature - MD5 - TripleDES - 1024 bits
  (b) RSA-Signature - SHA - TripleDES - 1024 bits
- IPSEC:
  (a) None - MD5, TripleDES - None
  (b) None - SHA, TripleDES - None
* Perfect Forward Secrecy: 1024 bits

2. PGPnet Version 7.0.3
   - Verläuft analog zu Version 6.5.8 (kleinere Unterschiede siehe Screenshots-Dokument)

1.5.3 Hinzufügen einer neuen Verbindung

- Unter Hosts - Add einen beliebigen Namen für die neue Verbindung vergeben.
- IP-Adresse, des FreeS/WAN Servers eintrage
- Secure Host wählen
- keine Shared Passphrase!
- Als Remote Athentication den PGP Key (public) des FreeS/WAN Servers wählen