Ipsec unter Linux 2.6

Einleitung:

Die native IPsec Implementierung im Linux Kernel ab Version 2.5.47 basiert auf dem USAGI Projekt. Hierbei handelt es sich um eine alternative Implementierung des IPsec Stacks zum FreeS/WAN Projekt. Daraus resultieren zunächst unterschiedliche Fähigkeiten und Anwendungsmöglichkeiten. Die Werkzeuge zur Administration unterscheiden sich von FreeS/WAN.

Mit dem FreeS/WAN Projekt existiert bereits seit 1996 ein Projekt, das die Implementierung eines IPsec Netzwerkstacks vorantreibt. Dennoch wurde von Dave Miller und Alexey Kuznetsov im Herbst 2002 eine native IPsec Implementierung basierend auf dem USAGI Projekt (http://www.linux-ipv6.org/) begonnen. Das USAGI Projekt versucht einen kompletten IPv6 Netzwerk Stack für Linux zu erzeugen. Die momentan im Kernel existierende IPv6 Implementierung ist unvollständig, sehr alt und fehlerhaft. Hierbei arbeitet das USAGI Projekt eng mit dem WIDE Projekt (http://www.wide.ad.jp/), dem TAHI Projekt (http://www.tahi.org/) und dem KAME Projekt (http://www.kame.net/) zusammen. Alle drei Projekte beschäftigen sich mit der Implementierung und dem Test von Ipv6.

IPSec kurz umrissen:

Es gibt grundsätzlich mehrere Möglichkeiten um eine IPSec Verbindung aufzubauen:

1. **Bietet Ipsec mehrere verschiedene Protokolle.**

   - AH -> stellt nur sicher, dass die Datenpakete nicht verändert wurden (nur im Transportmodus einsetzbar).

   - ESP -> Verschlüsselung der Leitung (im Transportmodus wie im Tunnelmodus einsetzbar)

2. **Gibt es mehrere Möglickkeiten des Verbindungsaufbaues:**

   - Preshared **Keys** (Key's um die Verbindung zu sichern werden händisch vor dem Aufbauen der Verbindung ausgetauscht)

   - Austausch der Schlüssel (x.509) über einen IKE Dämon (racoon, pluto, ...)

3. **Es gibt noch 2 verschiedene Transportmodien:**

   - TransportModus (zur Verbindung zweier Rechner direkt)

   - Tunnel Modus (kann auch verschiedene Netzwerke miteinandere verbinden)

4. **Besondere Targets:**

   - Komprimierung (unter Linux 2.6 wird seit neuestem auch eine Komprimierung der Ipsec Verbindung unterstützt).

Konfiguration für Linux 2.6

Networking support (NET) [Y/n/?] y
  *
  * Networking options
  *
  PF_KEY sockets (NET_KEY) [Y/n/m/?] y
  IP: AH transformation (INET_AH) [Y/n/m/?] y
  IP: ESP transformation (INET_ESP) [Y/n/m/?] y
  IP: IPsec user configuration interface (XFRM_USER) [Y/n/m/?] y

Cryptographic API (CRYPTO) [Y/n/?] y
  HMAC support (CRYPTO_HMAC) [Y/n/?] y
  Null algorithms (CRYPTO_NULL) [Y/n/m/?] y
  MD5 digest algorithm (CRYPTO_MD5) [Y/n/m/?] y
  SHA1 digest algorithm (CRYPTO_SHA1) [Y/n/m/?] y
  DES and Triple DES EDE cipher algorithms (CRYPTO_DES) [Y/n/m/?] y
  AES cipher algorithms (CRYPTO_AES) [Y/n/m/?] y

Userspace Configuration:

IKE Dämons:

Es können mit dem KAME Ipsec Stack in Linux 2.6 inzwischen mehrere verschiedene IKE Dämons zusammenarbeiten.

- isakmpd (openbsd)
- racoon (KAME)
- pluto (FreeS/WAN)

SETKEY:

```
spdadd 192.168.1.2 192.168.1.4 any -P in ipsec
        esp/transport//require
        ah/transport//require;

spdadd 192.168.1.4 192.168.1.2 any -P out ipsec
        esp/transport//require
        ah/transport//require;

spdadd 0.0.0.0/0 192.168.1.4/24  any -P in ipsec
        esp/tunnel/192.168.1.2-192.168.1.4/require;
spdadd 192.168.1.4/24 0.0.0.0/0 any -P out ipsec
        esp/tunnel/192.168.1.4-192.168.1.2/require;
```

Erzeugen der Zertifikate:

Je nach Distribution liegt das Script zum vereinfachten Erstellen der CA und Zertifikate in einem anderen Pfad.

Unter Debian liegt es unter /usr/lib/ssl/misc und heißt CA.sh/CA.pl

Erstellen der root-CA:

cd /usr/lib/ssl/misc

./CA.sh -newca

mit diesem Befehl wird das root-CA unter demoCA angelegt (private Schlüssel zum signieren und root-CA zertifikat)

Standardmäßig hat das Zertifikat unter Debian nur eine Gültigkeit für 356Tage, man sollte dies um einiges verlängern (in diesem Beispiel auf 10Jahre).

cd DemoCA

openssl x509 -in cacert.pem -days 3650 -out cacert.pem -signkey ./ private/cakey.pem

Erstellen der Client Zertifikate:

Linux: schritte 1, 2 und 3

Windows: scrhitte 1, 2 und 4

1.Erstellen des zertifikates

./CA.sh -newreq

2.Signieren des Zertifikates mit dem Root-CA

./CA.sh -sign

3.Entfernen der Passphrase von den Zertifikaten

openssl rsa -in newreq.pem -out newreq.pem

4.Umwandeln des Zertifikate für Windows2k/XP

openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -certfile demoCA/cacert.pem -out win-cert.p12

Installieren der Zertifikate unter Linux:

cd /usr/lib/ssl/misc

mkdir /etc/racoon/certs

cp newreq.pem newcert.pem demoCA/cacert.pem /etc/racoon/certs

Installieren der Zertifikate unter Windows:

<http://support.real-time.com/open-source/ipsec/index.html>

KAME Project: <http://www.kame.net/>

KAME Racoon: <http://www.kame.net/racoon/>

Ipsec on WinXP/2k: <http://vpn.ebootis.de/>

FreeBSD Ipsec Handbook: <http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ipsec.html>

Alles zusammengefasst über Ipsec unter Linux und auch allgemein: <http://einsteinmg.dyndns.org/projects/ipsec/>