Internet-Security mit IPsec – Haken und Ösen beim praktischen Einsatz

Hans-Jörg Höxer <Hans-Joerg.Hoexer@rommelwood.de>

26. November 2000

1 – Überblick

- Überblick über das IP Security Protocol
- IPsec mit OpenBSD
  - Konfiguration
- IPsec mit Linux
  - Installation
  - Konfiguration

2 – Sicherheitsprobleme bei IPv4

- Das Internetprotokoll bietet keinerlei Schutz für die übertragenen Daten (außer Fehlerkorrektur)
- Absender/Empfänger werden nicht authentifiziert.

3 – Anforderungen an Kommunikation

- Vertraulichkeit (Verschlüsselung)
- Integrität der Daten
- Authentifizierung (Signierung)
- Schutz vor Replay

4 – Das IPsec Protokoll

IPsec bietet zwei zusätzliche Protokolle an:

- Encapsulating Security Payload ESP
- Authentication Header AH

5 – ESP (IANA 50)

- bietet Verschlüsselung, Integritätsprüfung, Authentifizierung und Replayschutz für die **Daten** des IP-Pakets
- Verschlüsselung kann alleine verwendet werden
- Authentifizierung und Integritätsprüfung können alleine verwendet werden
- für Replayschutz wir Integritätsprüfung und Authentifizierung benötigt

6 – AH (IANA 51)

- bietet Integritätsprüfung, Authentifizierung und Replayschutz
- keine Verschlüsselung
- schützt im Gegensatz zu ESP auch den IP-Header selbst (Sender-/Empfängeradresse)

7 – Security Association (SA)

Parameter für ESP und AH werden durch eine Security Association beschrieben

- Verschlüsselungsalgorithmus
- Hashalgorithmus (für Authentifizierung)
- Schlüssel
- Gültigkeitslänge (Zeit, Volumen)
- etc.

8 – Securtiy Parameter Index (SPI)

Eine SA wird durch folgendes Triplet identifiziert:

- Zieladresse
- Security Parameter Index
- Protokolltyp

9 – Transportmode vs. Tunnelmode (1)

- Transportmode
  - Host to Host
  - Kombination von AH und ESP möglich
- Tunnelmode
  - zwischen Gateways
  - Kapselung von IP-Paketen in IP-Pakete
  - Bildung von Virtual Private Networks
  - keine Kombination von AH und ESP möglich

10 – Transportmode vs. Tunnelmode (2)

- normales IP-Pakete (hier TCP)

  `[IP header][TCP header][data]`

- ESP im Transportmode

  `[IP header][ESP header][TCP header][data]`

- ESP im Tunnelmode

  `[IP header][ESP header][IP header][TCP header][data]`

11 – Keymanagement

Schlüssel kann manuell oder automatisch mittels Dämonen erfolgen:

- IKE (Internet Key Exchange) mit ISAKMPD (Internet Security Association and Key Management Daemon) RFC 1408
- Photuris RFC 2522

12 – Ablauf von IKE

- Phase 1: Einrichtung einer SA, über die die Dämonen miteinander kommunizieren (Main- oder Aggressive-Mode)
- Phase 2: Aushandeln von SAs für IPsec (Quick-Mode)

13 – Szenario

- yerbouti 192.168.2.1
- redhat 192.168.2.10
- Verbindung Host-to-Host im Tunnelmode
- Main-mode: 3DES und SHA
- Quick-mode: 3DES und SHA (mit PFS), 3DES und MD5 (ohne PFS)

14 – IPsec mit OpenBSD

OpenBSD bietet vollständige Unterstützung von IPsec:

- ISAKMPD und Photuris
- viele verschiedene Algorithmen (DES, 3DES, Blowfish, AES, MD5, SHA, RIPEMD160, ...)
- Unterstützung von X509
- Kommunikation mit Kernel über PF_KEY-Socket

15 – Wichtige Programme und Dateien

- /sys/arch/<ARCH>/conf/<MACHINE>
- /sbin/isakmpd (8)
- /etc/isakmpd/isakmpd.conf (5)
- /etc/isakmpd/isakmpd.policy (5)
- /sbin/ipsecadm (8)
- /etc/rc.conf (8)
- /usr/sbin/sysctl (8)
- /etc/sysctl.conf (5)
- /sbin/ipf (8)
- /etc/ipf.rules (5)

16 – Konfiguration 2 – Kernel

- Kernel konfigurieren

```
option     CRYPTO              # Cryptographic Framework
option     IPSEC               # IPSEC VPN
#option    KEY                 # KEY implied by IPSEC
pseudo-device enc 4            # Encapsulation device
```

17 – Konfiguration 3 – sysctl

Mittel `sysctl` AH und ESP, evtl. IP-forwarding einschalten:

```
sysctl -w net.inet.esp.enable=1
sysctl -w net.inet.ah.enable=1

sysctl -w net.inet.ip.forwarding=1
```

18 – Konfiguration 4 – Firewall

Firewall (falls vorhanden) muss IPsec-Pakete und udp/500 Pakete durchlassen:

```
pass in proto esp from 192.168.2.10/32 to 192.168.2.1/32
pass out proto esp from 192.168.2.1/32 to 192.168.2.10/32

pass in on enc0 from 192.168.2.10/32 to 192.168.2.1/32
pass out on enc0 from 192.168.2.1/32 to 192.168.2.10/32

pass in on ne0 proto udp from 192.168.2.10/32 port=500 \
                          to 192.168.2.10/32 port=500
pass out on ne0 proto udp from 192.168.2.1/32 port=500 \
                          to 192.168.2.10/32 port=500
```

19 – Konfiguration 5 – isakmpd.conf

```
[General]
Listen-on=                  192.168.2.1

[Phase 1]
192.168.2.10=               ISAKMP-peer-redhat

[Phase 2]
Connections=                IPsec-redhat-yerbouti

[ISAKMP-peer-redhat]
Phase=                      1
Local-address=              192.168.2.1
Address=                    192.168.2.10
Configuration=              Default-main-mode
Authentication=             <shared secret>
```

```
[IPsec-redhat-yerbouti]
Phase=                  2
ISAKMP-peer=            ISAKMP-peer-redhat
Configuration=          Default-quick-mode
Local-ID=               Net-yerbouti
Remote-ID=              Net-redhat

[Net-yerbouti]
ID-type=                IPV4_ADDR_SUBNET
Network=                192.168.1.1
Netmask=                255.255.255.255

[Net-redhat]
ID-type=                IPV4_ADDR_SUBNET
Network=                192.168.2.10
Netmask=                255.255.255.255
```

```
[Default-main-mode]
EXCHANGE_TYPE=          ID_PROT
Transforms=             3DES-SHA

[Default-quick-mode]
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-SHA-PFS-SUITE, \
                        QM-ESP-3DES-MD5-SUITE
```

20 – Konfiguration 6 – isakmpd.policy

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: "passphrase:<shared secret>"
Conditions: app_domain == "IPsec policy" && \
            esp_present == "yes" -> "true";
```

21 – Konfiguration 7 – isakmpd starten

- `ipsecadm flush`
- `isakmpd -d -DA=<n>`
- `netstat -rn`
- in `rc.conf`:

  `isakmpd_flags=""`

22 – IPsec mit Linux FreeS/WAN

FreeS/WAN für Linux bietet ausreichende (d. h. minimale) IPsec-Unterstützung:

- Pluto für IKE, Photuris (experimentel)
- 3DES, MD5 und SHA1
- für X509 ist Patch nötig
- eigenes Format für PKI
- PF_KEY-Socket (teilweise)

23 – Konfiguration 1 – Sourcen und Kernel

- FreeS/WAN-Sourcen unter
  `ftp://ftp.xs4all.nl/pub/crypto/freeswan`
- akutelle Version ist 1.7
- Linuxkernel: 2.2.x, 2.3.x, 2.4.x

24 – Konfiguration 2 – Kompilieren

- Sourcen unter `/usr/src/linux` und `/usr/src/freeswan-1.7`
- erst normalen Kernel bauen
- dann FreeS/WAN mit `make menugo` und `make` (als root)
- Reboot

25 – Konfiguration 3 – sysctl und Firewall

Über proc-Dateisystem rp_filter ausschalten(!), evtl. Forwarding aktivieren, Firewall anpassen:

```
echo "0" > /proc/sys/net/ipv4/conf/<interface>/rp_filter
echo "1" > /proc/sys/net/ipv4/ip_forward

ipchains -A input -p esp -i eth0 -j ACCEPT
```

26 – Konfiguration 4 – ipsec.conf

```
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        plutowait=no

conn yerbouti-redhat
        auto=add
        type=tunnel
        left=192.168.2.1
        # leftsubnet=
        # leftnexthop=
```

```
right=192.168.2.10
# rightsubnet=
# rightnexthop=

keyexchange=ike
ikelifetime=1h
keylife=8h
keyingtries=5
pfs=yes
rekeymargin=9m
rekeyfuzz=25%
```

27 – Konfiguration 5 – ipsec.secrets

`ipsec.conf` darf nur für root lesbar sein!

```
192.168.2.1 192.168.2.10: PSK "<secret>"
```

`<secret>` kann mit `ipsec ranbits <x>` erzeugt werden.

28 – Konfiguration 6 – IPsec starten

- `ipsec spi --clear && ipsec eroute --clear`
- `ipsec look`
- IPsec wird über `/etc/init.d/ipsec {start|stop}` gesteuert.
- Test: `ping -p feedfacedeadbeef 192.168.2.10`
- Mit `tcpdump -x -e -i eth0` sieht man den verschlüsselten Traffic; mit `tcpdump -x -e -i ipsec0` sind die ursprünglichen IP-Pakete sichtbar.

29 – Links

- http://www.openbsd.org/faq/faq13.html
- http://www.freeswan.org/doc.html
- http://www.rommel.stw.uni-erlangen.de/\
  ~hshoexer/ipsec-howto/HOWTO.html