IPsec HOWTO

Ralf Spenneberg, ralf (at) spenneberg.net
2003-08-18

Do českého jazyka přeložil: Ivan Daler, ivan.daler (at) autocont.cz, 3.4.2006

Historie revizí

Revize 0.9.95	2005-09-03	Revidoval: RS
Přidáno nastavení MSS pomocí iptables pravidel a jedna menší oprava		
Revize 0.9.94	2005-07-19	Revidoval: RS
Přidány nějaké poznámky o směrování		
Revize 0.9.93	2005-03-03	Revidoval: RS
Opravena fwd politika, přidáno p12		
Revize 0.9.92	2005-02-01	Revidoval: RS
Přidána fwd politika		
Revize 0.9.91	2005-01-31	Revidoval: RS
/etc/setkey.conf nahradilo /etc/ipsec.conf		
Revize 0.9.9	2004-12-22	Revidoval: RS
Přidán NAT-Traversal. Změněna struktura dokumentu.		
Revize 0.9.6	2004-01-28	Revidoval: RS
Oprava modp768		
Revize 0.9.5	2004-01-08	Revidoval: RS
Přidána kompilace certpatch a keyconv		
Revize 0.9.4	2003-08-28	Revidoval: RS
Opravy		
Revize 0.9.3	2003-08-22	Revidoval: RS
Opravena typografie		
Revize 0.9.2	2003-08-19	Revidoval: RS
Opravena typografie		
Revize 0.9.1	2003-08-18	Revidoval: RS
Méně významné opravy		
Revize 0.9.0	2003-08-15	Revidoval: RS
Přidáno: Použití OpenBSD isakmpd		
Revize 0.8.3	2003-05-13	Revidoval: RS
Další typografické opravy. Přepracovány některé věty.		
Revize 0.8.2	2003-05-03	Revidoval: RS
Opravy chyb		
Revize 0.8.1	2003-04-30	Revidoval: RS
Přidána kapitola o certifikátech		
Revize 0.8	2003-04-18	Revidoval: RS
Hrubý návrh		

Abstrakt

Toto HowTo obsahuje základní a pokročilé kroky pro nastavení VPN s využitím IPsec technologie založené na linuxovém jádru 2.6. Poněvadž je k dispozici obrovské množství dokumentace pro linuxové jádro 2.4, soustředí se toto HowTo na nové rysy technologie IPsec v linuxovém jádře 2.6.

Obsah

Úvod

Nejnovější verze tohoto dokumentu vždy může být nalezena na stránkách Projektu linuxové dokumentace (The Linux Documentation Project[1]) a oficiální domovské stránce http://www.ipsec-howto.org.

Důvody pro psaní tohoto HowTo

V minulosti jsem užíval četná HowTo.Většina z nich pro mě byla velice cenná. Když v linuxovém jádru byly implementovány nové rysy technologie IPsec, začal jsem si s nimi hrát a používat je. Brzy jsem však zjistil, že existuje pouze velmi málo dokumentace. To způsobilo, že jsem začal psát toto HowTo.

Formát dokumentu

Dokument se dělí do sedmi kapitol.

1. kapitola: Úvod
   Tato kapitola.
2. kapitola: Teorie
   IPsec teorie. V podstatě IPsec protokoly.
3. kapitola: Openswan
   Tato kapitola bude popisovat jak nastavit Openswan na jádře 2.6.
4. kapitola: Racoon provozovaný na linuxovém jádře 2.6
   Kapitola popisuje jak nastavit IPsec VPN s využitím KAME nástrojů **setkey** a **racoon**. Také nyní zahrnuje NAT-Traversal.
5. kapitola: Isakmpd provozovaný na linuxovém jádře 2.6
   Kapitola popisuje jak nastavit IPsec VPN s využitím OpenBSD isakmpd IKE démonu.
6. kapitola: Tvorba X.509 certifikátů
   Kapitola popisuje jak generovat X.509 certifikáty pomocí příkazu **openssl**.
7. kapitola: Pokročilá konfigurace
   Kapitola uvádí nějaké pokyny pro XAUTH a pro užitečná **iptables** pravidla.

Přispěvatelé do dokumentu

- Matija Nalis
- Fridtjof Busse
- Uwe Beck
- Juanjo Ciarlante
- Ervin Hegedus
- Barabara Kane
- Alois Schmid

Právní informace

Copyright

Související dokumenty

Networking Overview HOWTO[3]
Networking HOWTO[4]
VPN-Masquerade HOWTO[5]
VPN HOWTO[6]
Advanced Routing & Traffic Control HOWTO[7]

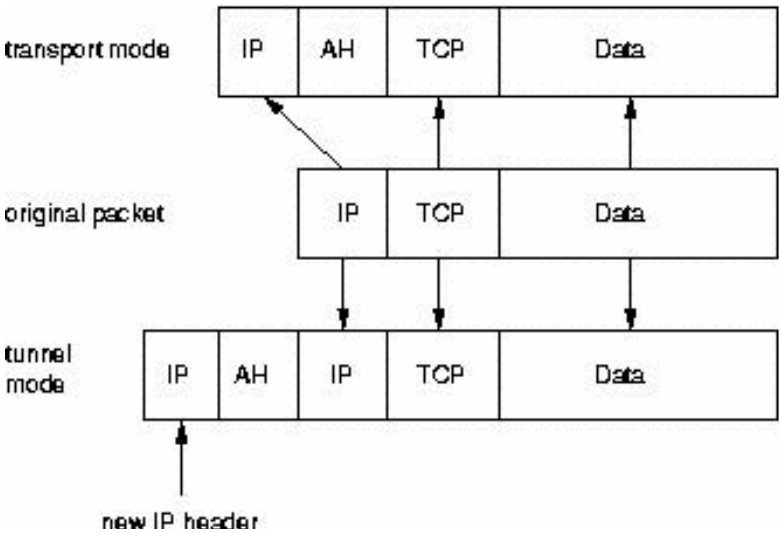
Teorie

Co je IPsec?

IPsec je rozšíření IP protokolu, které poskytuje bezpečnost pro IP protokol a protokoly vyšších vrstev. Nejdříve se vyvinul pro nový standard IPv6 a následně byl zpětně implementován na IPv4. IPsec architektura je popsaná v RFC2401. Několik následujících odstavců poskytuje krátký úvod do IPsec technologie.

IPsec užívá dva rozdílné protokoly - AH a ESP - aby zajistil ověřování (prokazování) identity (authentication), neporušenost (integrity) a důvěryhodnost (confidentiality) komunikace. Může chránit buď celý IP datagram nebo pouze protokoly vyšší vrstvy. Příslušné módy se nazývají tunelový (tunnel) mód a transportní (transport) mód. V tunelovém módu je IP datagram plně zapouzdřený do nového IP datagramu, který používá IPsec protokol. V transportním módu je pouze užitečná část (payload) IP datagramu zpracovaná IPsec protokolem tím způsobem, že se vkládá IPsec hlavička mezi IP hlavičku a hlavičku protokolu vyšší vrstvy (viz obrázek 1).

Obrázek 1. Tunelový a transportní mód technologie IPsec



Pro ochranu neporušenosti (integrity) IP datagramů IPsec protokoly používají HMAC (Hash Message Authentication Code). Aby IPsec protokoly získaly tento HMAC, užívají hash algoritmy jako MD5 a SHA a vypočtou hash na základě tajného klíče (secret key) a obsahu IP datagramu. Tento HMAC je potom zahrnut do hlavičky IPsec protokolu a příjemce paketu může kontrolovat HMAC, pokud má přístup k tajnému klíči.

Pro ochranu důvěryhodnosti (confidentiality) IP datagramů IPsec protokoly užívají standardní symetrické šifrovací (encryption) algoritmy. IPsec standard požaduje implementaci NULL a DES. Dnes jsou však obvykle užívané silnější algoritmy jako 3DES, AES a Blowfish.

Pro ochranu proti útokům typu DoS (Denial of Service) IPsec protokoly užívají tzv. okno (sliding window). Každému paketu je přiřazeno sekvenční číslo (Sequence Number) a je přijatý, pouze jestliže číslo paketu je v rámci daného okna nebo novější. Starší pakety jsou okamžitě zrušeny. Toto chrání proti útokům založeným na opakování paketů (replay attacks), kdy útočník zaznamená původní pakety a přehraje je později.

Aby protější strana komunikace (peer) mohla zapouzdřit (encapsulate) a rozpouzdřit (decapsulate) IPsec pakety, potřebuje nějaký způsob, jak uložit tajné klíče, algoritmy a IP adresy zahrnuté v komunikaci. Všechny tyto parametry potřebné pro ochranu IP datagramů jsou uložené v SA (Security Association). SAs (tj. mn. č. pro SA) jsou uloženy v databázi SAD (Security Association Database).

Každé SA definuje následující parametry:

- Zdrojovou a cílovou IP adresu výsledné IPsec hlavičky. Toto jsou IP adresy protějších stran IPsec komunikace chránících pakety.
- IPsec protokol (AH nebo ESP), někdy je také podporovaná komprese (IPCOMP).
- Algoritmus a tajný klíč užívaný IPsec protokolem.
- SPI (Security Parametr Index). Tj. 32 bitové číslo, které identifikuje SA.

Některé implementace SAD databáze umožňují, aby byly ukládány i další parametry:

- IPsec mód (tunelový nebo transportní)
- Velikost okna (sliding window) pro ochranu proti útokům založeným na přehrání zachycených dat.

- Doba SA existence.

Poněvadž SA definuje zdrojové a cílové IP adresy, může chránit jen jeden směr datového provozu (traffic) v plně duplexní IPsec komunikaci. Aby se chránily oba směry, IPsec potřebuje dva jednosměrné SAs.

SA pouze specifikuje jak IPsec chrání datový provoz. Další informace je potřebná pro definování toho, který datový provoz chránit. Tato informace je uložena v SP (Security Policy), což je uloženo v databázi SPD (Security Policy Database).

SP obvykle určuje následující parametry:

- Zdrojová a cílová adresa paketů, které se mají chránit. V transportním módu jsou to stejné adresy jako v SA. V tunelovém módu se mohou lišit!
- Protokol a port, jež se mají chránit. Některé IPsec implementace nedovolují definice specifických protokolů, které se mají chránit. V tomto případě je chráněn veškerý datový provoz mezi zmíněnými adresami.
- SA, které se má použít pro ochranu paketů.

Ruční nastavení SA je docela náchylné na chyby a není příliš bezpečné. Ve virtuální soukromé síti (VPN) musejí být tajné klíče a šifrovací algoritmy sdílené mezi všemi účastníky komunikace (peers). Kritické problémy pro systémové administrátory speciálně představuje výměna klíčů: Jak vyměnit tajné symetrické klíče, když ještě není žádné šifrování?

Aby se vyřešil tento problém, byl vyvinut IKE (Internet Key Exchange) protokol. Tento protokol v první fázi ověřuje identitu protějšků komunikace. Ve druhé fázi jsou vyjednané SAs a jsou vybrané tajné symetrické klíče pomocí výměny klíčů metodou Diffie Hellmana. IKE protokol se potom dokonce stará o periodickou změnu tajných klíčů, aby se zajistila důvěryhodnost.

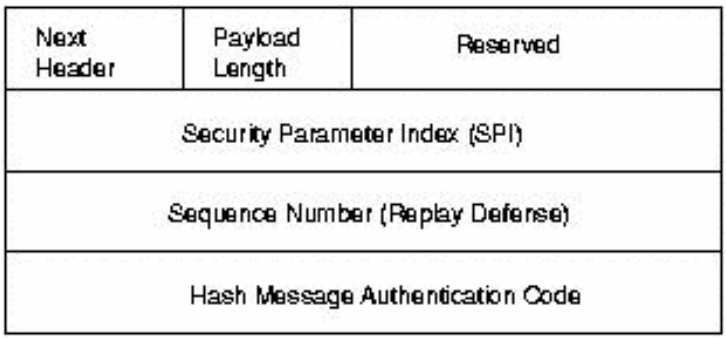
IPsec protokoly

Rodina IPsec protokolů se skládá ze dvou protokolů: AH (Authentication Header) a ESP (Encapsulated Security Payload). Oba jsou nezávislé protokoly. AH je IP protokol číslo 51 a ESP je protokol číslo 50 (viz */etc/protocols*). Následující dva odstavce se budou stručně zabývat jejich vlastnostmi.

AH - Authentication Header

AH protokol chrání integritu IP datagramu. Aby se toho dosáhlo, AH protokol počítá HMAC. Když se počítá HMAC, AH protokol vychází z tajného klíče, užitečné části (payload) paketu a neměnných částí IP hlavičky jako jsou IP adresy. Potom k paketu přidává AH hlavičku. AH hlavička je zobrazena na obrázku 2.

Obrázek 2. AH hlavička chrání integritu paketu



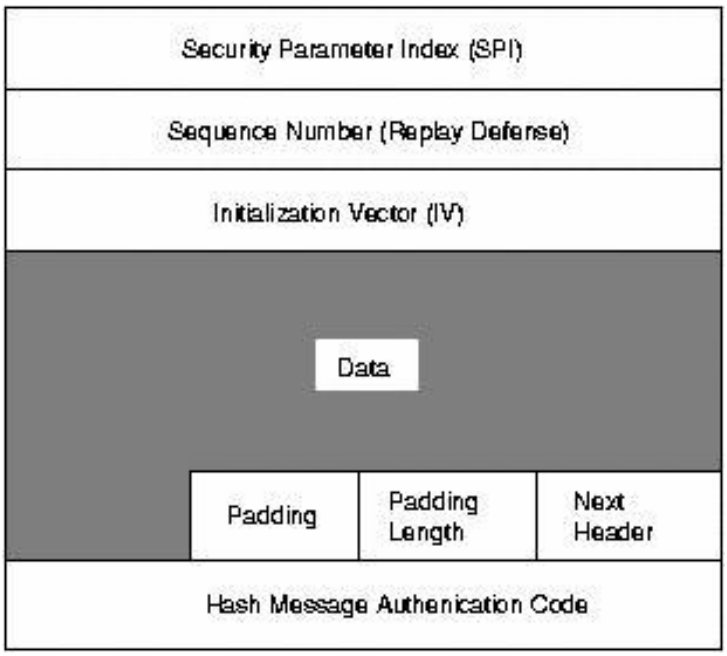
AH hlavička je 24 bajtů dlouhá. První bajt je políčko *Next Header*. Toto políčko určuje protokol následující hlavičky. V tunelovém módu je zapouzdřen celý IP datagram, proto hodnota tohoto políčka je 4. Když se zapouzdřuje TCP datagram v transportním módu, odpovídající hodnota je 6. Další bajt specifikuje délku užitečné části (payload) paketu. Toto políčko je nasledované dvěma rezervovanými bajty. Další dvojité slovo (double word) udává 32 bitů dlouhé SPI (*Security Parameter Index*). SPI specifikuje SA, které se má použít pro rozpouzdření paketu. 32 bitů dlouhé sekvenční číslo (*Sequence Number*) chrání proti útokům založeným na opětném přehrání zachycených dat. Konečně posledních 96 bitů obsahuje HMAC (*Hash Message Authentication Code*). HMAC chrání integritu paketů, poněvadž pouze protější strany komunikace znající tajný klíč mohou vytvořit a kontrolovat HMAC.

AH nedovoluje NAT, poněvadž AH protokol chrání IP datagram včetně neměnných částí IP hlavičky jako jsou IP adresy. NAT (Network Address Translation) nahrazuje IP adresu v IP hlavičce jinou (obvykle zdrojovou) IP adresou. Po této náhradě již není HMAC platný. Toto omezení obchází implementace rozšíření IPsec protokolu NAT-Traversal.

ESP - Encapsulated Security Payload

ESP protokol může zajistit jak integritu paketu pomocí HMAC, tak i důvěryhodnost pomocí šifrování. Po zašifrování paketu a výpočtu HMAC se vytvoří ESP hlavička a přidá se do paketu. ESP hlavička se skládá ze dvou částí a je prezentována na obrázku 3.

Obrázek 3. ESP hlavička



První dvojité slovo v ESP hlavičce specifikuje SPI (*Security Parameter Index*). SPI udává SA, jež se má použít pro rozpouzdření ESP paketu. Druhé dvojité slovo obsahuje sekvenční číslo (*Sequence Number*). Toto sekvenční číslo chrání proti útokům založeným na opětném přehrání zachycených dat. Třetí dvojité slovo určuje IV (*Initialization Vector*), který je používán v šifrovacím procesu. Symetrické šifrovací algoritmy jsou náchylné k frekvenčnímu útoku (frequency attack), pokud není užíván žádný IV. IV zajišťuje, že dvě stejné užitečné části paketů vedou ke dvěma různým zašifrovaným užitečným částem paketů.

IPsec pro šifrovací proces používá blokové šifry. Jestliže délka užitečné části není násobkem délky bloku, může být potřeba užitečnou část paketu vyplnit "bajtovou vycpávkou" (pad). Potom je přidaná délka vycpávky. Za délkou vycpávky následuje *Next Header* o délce 2 bajty, který specifikuje následující hlavičku. Nakonec je do ESP hlavičky přidán 96 bitů dlouhý HMAC zajišťující integritu paketu. HMAC bere do úvahy pouze užitečnou část paketu. IP hlavička není zahrnuta do výpočetního procesu.

Použití překladu adres NAT proto neporuší ESP protokol. Ve většině případů však stále ještě není NAT možný ve spojení s IPsec technologií. V tomto případě NAT-Traversal nabízí řešení zapouzdřením ESP paketů do UDP paketů.

IKE protokol

IKE protokol řeší nejvýznačnější problém v nastavení bezpečné komunikace: ověřování identity účastníků komunikace a výměnu symetrických klíčů. Potom tvoří SAs a naplňuje SAD. IKE protokol obvykle běží jako uživatelský démon a není implementován v operačním systému. IKE protokol užívá pro svoji komunikaci UDP port 500.

IKE protokol funguje ve dvou fázích. První fáze zřizuje ISAKMP SA (Internet Security Association and Key Management Protocol SA). Ve druhé fázi je ISAKMP SA užívané pro vyjednání a nastavení IPsec SAs.

Ověřování identity protějších stran komunikace v první fázi může být založeno na (před)sdílených klíčích (Pre-Shared Keys, PSK), RSA klíčích a X.509 certifikátech (**racoon** dokonce podporuje i Kerberos).

První fáze obvykle podporuje dva různé módy: hlavní (main) a agresivní (aggressive) mód. Oba módy sice ověřují identitu protějších stran komunikace a vytvářejí ISAKMP SA, ale agresivní mód užívá pouze polovinu zpráv, aby dosáhnul tohoto cíle. To má ale svoji stinnou stránku, protože argesivní mód nepodporuje zabezpečení procesu ověření identity a je proto náchylný na útok typu "muž uprostřed" (man-in-the-middle attack), pokud je užívaný ve spojení se sdílenými klíči. Na druhou stranu je to jediný účel agresivního módu. Z důvodu vnitřního fungování hlavního módu, tento mód nepodporuje použití různých sdílených klíčů pro neznámé protější strany komunikace. Agresivní mód nepodporuje zabezpečení procesu ověření identity a přenáší identitu klienta otevřeným způsobem. Protější strany komunikace se proto musejí před procesem ověřování identity znát, aby se mohly použít různé sdílené klíče pro různé účastníky komunikace.

Ve druhé fázi IKE protokol vyměňuje SA návrhy a vyjednává SAs založené na ISAKMP SA. ISAKMP SA poskytuje ověřování identity jako ochranu proti útoku typu "muž uprostřed". Tato druhá fáze užívá rychlý mód (quick mode).

Obvykle dva účastníci komunikace vyjednávají pouze jeden ISAKMP SA, který je potom užívaný pro vyjednání několika (alespoň dvou) jednosměrných IPsec SAs.

NAT-Traversal

Co je NAT-Traversal a proč je potřebný?

Často je jedna strana VPN za nějakým zařízením provádějícím NAT. Zde předpokládám, že jde o SNAT (Source NAT). Kdykoliv hovořím o NAT, mám na mysli SNAT nebo maškarádu (Masquerading). Co to znamená ve vztahu k VPN? Nuže, především původní IP adresa jedné strany komunikace je skryta zařízením provádějícím NAT. NAT zařízení skryje původní IP adresu a nahradí ji svoji vlastní IP adresou.

To dělá IPsec AH protokol okamžitě nepoužitelný. Ale ESP se stále ještě může používat, jestliže jsou obě strany komunikace správně konfigurované.

Tak proč potřebujeme NAT-Traversal? Protože jakmile dva počítače za stejným NAT zařízením zkoušejí vytvořit tunel směrem ven, v obou případech to selže.

Proč se tak stane? NAT zařízení potřebuje zaznamenat informaci o "natovaných" spojeních, aby mohlo "denatovat" pakety odpovědí směřujících zpět k původnímu klientovi. Proto NAT zařízení udržuje vnitřní tabulku, kde jsou uloženy veškeré informace o "natovaných" spojeních. Předpokládejme, že se jeden klient připojuje k webovému serveru na Internetu. NAT zařízení skryje původní adresu a nahradí ji svoji vlastní. Potom udělá poznámku ve své vnitřní tabulce, že všechny pakety vracející se zpět na vybraný klientský port se musejí posílat k původnímu klientovi. Jakmile druhý klient zahájí spojení, NAT zařízení s ním nakládá stejně. Jestliže druhý klient náhodnou zvolil stejný klientský port, NAT zařízení bude pro jednoznačnost modifikovat také klientský port. Toto pracuje velmi dobře pro TCP a UDP, protože tyto protokoly poskytují porty. ESP však porty neužívá. Proto NAT zařízení pouze může používat protokol rozlišující porty. Když navazuje spojení první klient, NAT zařízení ukládá informaci do tabulky, že všechny ESP pakety musí být "denatované" k prvnímu klientovi. Když se spojuje druhý klient, přepíše tento záznam v tabulce příslušnou položkou pro druhého klienta a tak rozbije minimálně první spojení.

Čím NAT-Traversal pomáhá? NAT-Traversal opět zapouzdřuje ESP pakety do UDP paketů. Ty mohou být snadno zpracované NAT zařízením, poněvadž poskytují porty. Ve výchozím (default) nastavení je užívaný UDP port 4500. NAT-Traversal je specifikován v několika předlohách standardů (drafts). Nyní nejsou žádná RFC. Pěkným rysem technologie NAT-Traversal je fakt, že jakmile je aktivován, užívá se, je-li potřeba, automaticky.

Openswan provozovaný na operačním systému Linux, verze jádra 2.6

Určeno k doplnění.

Linuxové jádro 2.6 užívající nástroje KAME

Tato kapitola vysvětluje užití IPsec technologie linuxového jádra >= 2.5.47 a 2.6.*. Instalace a nastavení této IPsec implementace se značně liší od FreeS/WAN a je podobné *BSD variantám jako FreeBSD, NetBSD a OpenBSD.

Nejdříve se budu zabývat konfigurací a instalací linuxového jádra a uživatelských nástrojů (user space tools). Potom bude vysvětleno nastavení spojení v transportním a tunelovém módu, pokud jsou parametry pro spojení tvořeny ručně (manually keyed). Nakonec se budeme zabývat nastavením spojení se sdílenými klíči a X.509 certifikáty, pokud jsou parametry spojení generovány automaticky (automatically keyed). Podpora cestujících uživatelů (roadwarriors) bude vysvětlena jako poslední.

Instalace

Instalace žádá linuxové jádro alespoň verze 2.5.47 nebo 2.6.*. Zdrojový kód jádra může být stažen z http://www.kernel.org. Po stažení zdrojového kódu balíček zdrojového kódu jádra musí být rozbalen, konfigurován a kompilován.

```
cd /usr/local/src
tar xvjf /path-to-source/linux-<version>.tar.bz2
cd linux-<version>
make xconfig
make bzImage
make modules
make modules_install
make install
```

Toto jsou nejčastěji užívané příkazy pro konfiguraci a kompilaci linuxového jádra. Jestliže potřebujete speciální nastavení, laskavě se odkažte k dokumentaci Kernel-HowTo.

Když konfigurujete jádro, je důležité zapnout následující položky:

```
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
```

V závislosti na verzi užívaného jádra snad budete muset zapnout i podporu IPv6.

Jakmile je jádro zkompilované a nainstalované, mohou se instalovat uživatelské nástroje. V současnosti jsou tyto nástroje udržované na http://ipsec-tools.sourceforge.net/. Když budete kompilovat balíček ručně, bude možná potřeba určit umístění hlaviček jádra (kernel headers). Tento balíček potřebuje hlavičky jádra minimálně verze 2.5.47.

Upozornění: Když užíváte linuxové jádro >= 2.6.10, musíte použít ipsec-tools >= 0.5, protože toto jádro přidává novou politiku směrování (forward policy) neznámou démonu racoon ve starších verzích ipsec-tools. Buďte si vědomi, že některé linuxové distribuce hodně záplatují dokonce i starší linuxová jádra, což vás může také postihnout. Jen zkontrolujte **fwd** politiky ve vašem jádře.

```
./configure --with-kernel-headers=/lib/modules/2.6.X/build/include
make
make install
```

Nyní by vše mělo být připraveno.

Ručně nastavované parametry spojení s pomocí setkey

Formulace "ručně nastavované parametry spojení" znamená, že veškeré parametry potřebné pro nastavení spojení poskytne administrátor. IKE protokol není užíván, aby automaticky ověřoval identitu účastníků spojení a vyjednával tyto parametry. Administrátor rozhoduje jaký protokol, algoritmus a klíč se mají použít pro vytvoření SAs a podle toho naplní SAD.

Transportní mód

Tento odstavec se bude nejdříve zabývat ručním nastavením parametrů spojení v transportním módu. To je pravděpodobně nejlepší spůsob jak začít, protože z hlediska nastavení se jedná o nejjednodušší spojení. V tomto odstavci se předpokladá, že dva počítače s adresami 192.168.1.100 a 192.168.2.100 komunikují s využitím technologie IPsec.

Veškeré parametry uložené v databázích SAD a SPD mohou být modifikované pomocí příkazu **setkey**. Tento příkaz má docela důkladnou stránku dokumentace man. Proto zde jsou obsažené pouze volby (options) potřebné pro nastavení spojení v transportním módu. **setkey** čte své příkazy ze souboru, pokud je volán ve tvaru **setkey -f /etc/setkey.conf**. Vhodný soubor */etc/setkey.conf* je ukázán v následujícím výpisu.

```
#!/usr/sbin/setkey -f

# Configuration for 192.168.1.100

# Flush the SAD and SPD
flush;
spdflush;

# Attention: Use this keys only for testing purposes!
# Generate your own keys!

# AH SAs using 128 bit long keys
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd 192.168.1.100 192.168.2.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P in ipsec
```

```
        esp/transport//require
        ah/transport//require;
```

Pokud chcete použít spojení s ručně tvořenými klíči jinak než pro testovací účely, budete potřebovat nějaké klíče, aby jste nahradili klíče v tomto skriptu. Aby jste vytvořili vaše klíče, užijte příkaz podobný následujícímu:

```
# 128 Bit long key
$ dd if=/dev/random count=16 bs=1| xxd -ps
16+0 Records in
16+0 Records out
cd0456eff95c5529ea9e918043e19cbe

# 192 Bit long key
$ dd if=/dev/random count=24 bs=1| xxd -ps
24+0 Records in
24+0 Records out
9d6c4a8275ab12fbfdcaf01f0ba9dcfb5f424c878e97f888
```

Když tvoříte klíče, použijte prosím zařízení **/dev/random**, protože zajišťuje náhodné klíče.

Skript nejdříve vyčistí (flush) databáze SAD a SPD. Potom vytvoří AH SAs a ESP SAs. Příkaz **add** přídá SA do databáze SAD a vyžaduje zdrojovou a cílovou IP adresu, IPsec protokol (**ah**), SPI (**0x200**) a algoritmus. Algoritmus ověřování je specifikován volbou **-A** (šifrování pomocí **-E**, komprese je **-C**; IP komprese ještě není podporovaná). Po algoritmu musí být zadán klíč. Klíč může být zadán jako "ASCII" text uzavřený do dvojitých uvozovek nebo mu v hexadecimálním zápisu předchází **0x**.

Linux podporuje následující algoritmy pro AH a ESP: hmac-md5 a hmac-sha, des-cbc a 3des-cbc. Za krátký čas budou pravděpodobně podporované následující algoritmy: jednoduchý (bez šifrování), blowfish-cbc, aes-cbc, hmac-sha2-256 a hmac-sha2-512.

spdadd přidává bezpečnostní politiky do SPD. Tyto politiky definují jaké pakety budou chráněny technologií IPsec a jaké protokoly a klíče se mají použít. Příkaz vyžaduje zdrojové a cílové IP adresy paketů, které budou chráněny, protokol (a port) užívaný pro ochranu (any) a užívanou politiku (**-P**). Politika specifikuje směr (in/out), aplikovanou akci (ipsec/discard/none), protokol (ah/esp/ipcomp), mód (transport) a úroveň (use/require).

Konfigurační soubor musí být vytvořen na obou zúčastněných stranách IPsec komunikace. Zatímco prezentovaný výpis beze změny pracuje na straně 192.168.1.100, na straně 192.168.2.100 musí být mírně upraven, aby zohlednil změnu směru paketů. Nejjednodušší způsob jak to provést, je vyměnit směry v bezpečnostních politikách: nahraďte **-P in** s **-P out** a naopak. To je ukázané v následujícím výpisu:

```
#!/usr/sbin/setkey -f

# Configuration for 192.168.2.100

# Flush the SAD and SPD
flush;
spdflush;

# Attention: Use this keys only for testing purposes!
# Generate your own keys!
```

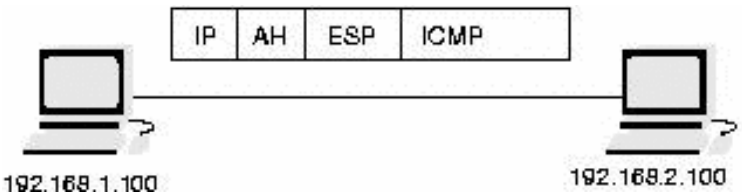
```
# AH SAs using 128 bit long keys
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd 192.168.1.100 192.168.2.100 any -P in ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;
```

Jakmile je konfigurační soubor umístěn na obou stranách komunikace, může se načíst příkazem **setkey -f /etc/setkey.conf**. Úspěšné vykonání příkazu lze otestovat zobrazením databází SAD a SPD:

```
# setkey -D
# setkey -DP
```

Nastavení nyní připomíná situaci na obrázku 4.

Obrázek 4. Dva počítače v transportním módu užívající AH a ESP



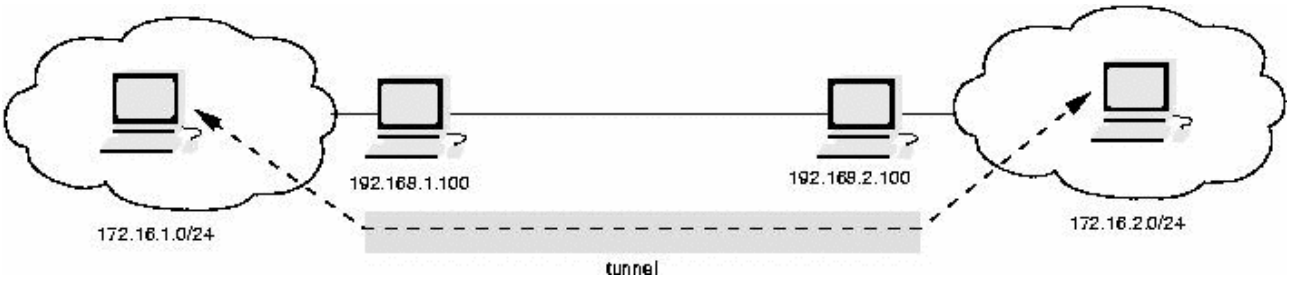
Jestliže nyní provedeme ping z jedné strany na druhou, datový provoz bude šifrovaný a tcpdump ukáže následující pakety:

```
12:45:39.373005 192.168.1.100 > 192.168.2.100: AH(spi=0x00000200,seq=0x1):
ESP(spi=0x00000201,seq=0x1) (DF)
12:45:39.448636 192.168.2.100 > 192.168.1.100: AH(spi=0x00000300,seq=0x1):
ESP(spi=0x00000301,seq=0x1)
12:45:40.542430 192.168.1.100 > 192.168.2.100: AH(spi=0x00000200,seq=0x2):
ESP(spi=0x00000201,seq=0x2) (DF)
12:45:40.569414 192.168.2.100 > 192.168.1.100: AH(spi=0x00000300,seq=0x2):
ESP(spi=0x00000301,seq=0x2)
```

Tunelový mód

Tunelový mód je používaný, když dvě zúčastněné strany komunikace využívající IPsec technologie pracují jako brány (gateways) a chrání datový provoz mezi dvěma sítěmi. (Obrázek 5). Původní IP pakety jsou šifrované a zapouzdřené jednou branou a přenesené k protější bráně. Protější brána rozpouzdří paket a předá originální nechráněný paket.

Obrázek 5. Dvě zúčastněné strany komunikace (brány) chrání datový provoz mezi dvěma sítěmi



Konfigurace SAs a politik tunelového módu je podobná transportnímu módu a je ukázaná v následujícím výpisu.

```
#!/usr/sbin/setkey -f

# Flush the SAD and SPD
flush;
spdflush;

# ESP SAs doing encryption using 192 bit long keys (168 + 24 parity)
# and authentication using 128 bit long keys
add 192.168.1.100 192.168.2.100 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.2.100 192.168.1.100 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Security policies
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
           esp/tunnel/192.168.1.100-192.168.2.100/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
           esp/tunnel/192.168.2.100-192.168.1.100/require;
```

Upozornění: Když užíváte linuxové jádro >= 2.6.10, musíte také, jestliže mají být pakety směrované počítačem, definovat politiku směrování (forward policy). Jen se přesvědčte, že užíváte ipsec-tools 0.5, které přidávají tuto politiku automaticky nebo ji přidejte sami, pokud používáte starší verzi nástrojů (ipsec-tools). Jestliže provozujete setkey v režimu Kernel-mode (-k), musíte také přidat fwd politiku ručně.

```
spdadd 172.16.2.0/24 172.16.1.0/24 any -P fwd ipsec
           esp/tunnel/192.168.2.100-192.168.1.100/require;
```

Tento příklad pouze užívá ESP protokol. ESP protokol může zajistit integritu a důvěryhodnost. V tomto případě je pořadí ESP algoritmů důležité. Nejdříve potřebujete definovat šifrovací algoritmus a jeho klíč a za druhé ověřovací algoritmus a jeho klíč.

Tento soubor musíte kopírovat na protější stranu tunelu a nahradit směr politik (**in** versus **out**). Jestliže užíváte politiku směrování, musíte navíc obrátit pořadí IP adres.

Na rozdíl od IPsec implementace na BSD, se může SA na Linuxu pouze používat buď v transportním módu nebo tunelovém módu. Transportní mód je výchozí, takže kdykoliv je požadován tunelový mód, SA se musí definovat pomocí **–m tunnel**.

Bezpečnostní politiky nyní určují IP adresy chráněných sítí. Pakety, které užívají tyto zdrojové a cílové IP adresy budou chráněné IPsec technologií. Kdykoliv se užívá tunelový mód, bezpečnostní politika musí specifikovat tunel a IP adresy skutečných konců tunelu (peers) uskutečňujících ochranu. Tato informace se potřebuje pro nalezení příslušného IPsec SA.

Jestliže tunel nepracuje, zkontrolujte vaše směrování (routing). Vaše počítače (hosts) potřebují vědět, že by měly posílat pakety směřující na protější síť na vaši VPN bránu. Nejjednodušším nastavením by bylo používat vaši VPN bránu jako výchozí (default) bránu.

Automaticky generované parametry spojení s pomocí démonu racoon

Na Linux byl také přenesen (ported) KAME IKE démon **racoon**. Tento démon je schopen automaticky nastavit IPsec spojení. **Racoon** podporuje ověřování (authentication) pomocí sdílených klíčů, X.509 certifikatů a Kerberosu. Démon může užívat hlavní mód, agresivní mód a základní (base) mód ve fázi jedna protokolu IKE. Tento odstavec ukáže konfiguraci démonu **racoon** v hlavním módu s využitím sdílených klíčů a X.509 certifikátů (zbývá doplnit Kerberos). Na závěr bude stručně vysvětlena konfigurace scénáře cestujícího uživatele (roadwarrior).

Nezapomeňte: Jestliže používáte ve vaší distribuci linuxové jádro 2.6.10 (nebo hodně záplatované 2.6.9), potřebujete ipsec-tools verze 0.5.

Sdílené klíče

Nejjednoduším způsobem ověřování identity pomocí démonu **racoon** je užití sdílených klíčů. Tyto klíče musí být definované v souboru */etc/psk.txt*. Tento soubor by neměl být čitelný pro neprivilegované uživatele (**chmod 400 /etc/psk.txt**) a má následující syntaxi:

```
# IPv4 Addresses
192.168.2.100          simple psk
5.0.0.1                0xe10bd52b0529b54aac97db63462850f3
# USER_FQDN
ralf@spenneberg.net    This is a psk for an email address
# FQDN
www.spenneberg.net     This is a psk
```

Soubor je organizován do sloupců. První sloupec obsahuje identitu účastníka komunikace ověřovaného pomocí PSK. Vše co začíná ve druhém sloupci je PSK.

Konfigurace démonu **racoon** je zřejmá. Následující výpis ukazuje typický konfigurační soubor */etc/racoon.conf*:

```
path pre_shared_key "/etc/psk.txt";

remote 192.168.2.100 {
        exchange_mode main;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method pre_shared_key;
```

```
                dh_group modp1024;
        }
}

sainfo address 172.16.1.0/24 any address 172.16.2.0/24 any {
        pfs_group modp768;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

Tento konfigurační soubor nejdříve definuje, kde může démon **racoon** nalézt sdílené klíče. Potom definuje protější stranu komunikace a parametry pro první fázi vyjednávání protokolu IKE. Druhý odstavec specifikuje parametry, které se mohou používat pro nastavení SAs. Tato definice může být specifická pro definované IP adresy nebo obecná, když se použije **anonymous** místo IP adres. Zde jsou pro SA definované šifrovací, ověřovácí a kompresní algoritmy. Definovat se musejí všechny tři, aby se vyhnulo chybě během startu démonu **racoon**.

IKE démon **racoon** nezahajuje vyjednávání parametrů tunelu okamžitě při startu. **Racoon** spíše čeká až je tunel potřebný. Aby se realizovalo příslušné upozornění, jádro potřebuje vědět, kdy **racoon** upozornit. Aby se toho dosáhlo, administrátor musí definovat bezpečnostní politiky bez příslušných SAs. Kdykoliv linuxové jádro potřebuje chránit paket podle bezpečnostních politik, přičemž není k dispozici žádné SA, linuxové jádro volá **racoon** a žádá o potřebné SAs. **Racoon** potom zahájí IKE vyjednávání a při jeho ukončení vytvoří SAs. Poté linuxové jádro může odeslat pakety.

Pro předpokládanou konfiguraci jsou na 192.168.1.100 potřebné následující politiky:

```
#!/usr/sbin/setkey -f
#
# Flush SAD and SPD
flush;
spdflush;

# Create policies for racoon
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
           esp/tunnel/192.168.1.100-192.168.2.100/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
           esp/tunnel/192.168.2.100-192.168.1.100/require;
```

Jakmile jsou politiky načtené pomocí příkazu **setkey –f /etc/setkey.conf**, může se startovat démon **racoon**. Pro testovací účely by se měl démon **racoon** startovat příkazem **racoon –F –f /etc/racoon.conf**. Konfigurace protější strany komunikace se zase musí modifikovat tak, aby zohlednila rozdílné směry. V souborech /etc/psk.txt, /etc/setkey.conf a /etc/racoon.conf se musí zaměnit IP adresy.

Inicializaci tunelu potom můžeme sledovat v log souboru:

```
2003-02-21 18:11:17: INFO: main.c:170:main(): @(#)racoon 20001216 20001216
 sakane@kame.net
2003-02-21 18:11:17: INFO: main.c:171:main(): @(#)This product linked Open
SSL 0.9.6b [engine] 9 Jul 2001 (http://www.openssl.org/)
2003-02-21 18:11:17: INFO: isakmp.c:1365:isakmp_open(): 127.0.0.1[500] use
d as isakmp port (fd=7)
2003-02-21 18:11:17: INFO: isakmp.c:1365:isakmp_open(): 192.168.1.100[500]
 used as isakmp port (fd=9)
```

```
2003-02-21 18:11:37: INFO: isakmp.c:1689:isakmp_post_acquire(): IPsec-SA r
equest for 192.168.2.100 queued due to no phase1 found.
2003-02-21 18:11:37: INFO: isakmp.c:794:isakmp_ph1begin_i(): initiate new
phase 1 negotiation: 192.168.1.100[500]<=>192.168.2.100[500]
2003-02-21 18:11:37: INFO: isakmp.c:799:isakmp_ph1begin_i(): begin Identit
y Protection mode.
2003-02-21 18:11:37: INFO: vendorid.c:128:check_vendorid(): received Vendor
 ID: KAME/racoon
2003-02-21 18:11:37: INFO: vendorid.c:128:check_vendorid(): received Vendor
 ID: KAME/racoon
2003-02-21 18:11:38: INFO: isakmp.c:2417:log_ph1established(): ISAKMP-SA es
tablished 192.168.1.100[500]-192.168.2.100[500] spi:6a01ea039be7bac2:bd288f
f60eed54d0
2003-02-21 18:11:39: INFO: isakmp.c:938:isakmp_ph2begin_i(): initiate new p
hase 2 negotiation: 192.168.1.100[0]<=>192.168.2.100[0]
2003-02-21 18:11:39: INFO: pfkey.c:1106:pk_recvupdate(): IPsec-SA establish
ed: ESP/Tunnel 192.168.2.100->192.168.1.100 spi=68291959(0x4120d77)
2003-02-21 18:11:39: INFO: pfkey.c:1318:pk_recvadd(): IPsec-SA established:
 ESP/Tunnel 192.168.1.100->192.168.2.100 spi=223693870(0xd554c2e)
```

X.509 certifikaty

Pro proces ověřování **racoon** podporuje použití X.509 certifikátů. Tyto certifikáty se mohou kontrolovat certifikační autoritou (Certificate Authority, CA). Konfigurace je podobná PSK konfiguraci a liší se pouze v ověřovácí části:

```
path certificate "/etc/certs";

remote 192.168.2.100 {
        exchange_mode main;
        certificate_type x509 "my_certificate.pem" "my_private_key.pem";
        verify_cert on;
        my_identifier asn1dn;
        peers_identifier asn1dn;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method rsasig;
                dh_group modp1024;
        }
}

sainfo address 172.16.1.0/24 any address 172.16.2.0/24 any {
        pfs_group modp768;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

Certifikát a soukromý klíč (private key) jsou uložené pod certifikační cestou */etc/certs*. Tato cesta je nastavena direktivou **path certificate** v konfiguračním souboru. Certifikáty a seznamy stornovaných certifikátů (Certificate Revocation Lists, CRLs) vytvořené příkazem **openssl** jsou uložené ve formátu PEM. Návod pro tvoření certifikátů je v kapitole o X.509 certifikátech. Jestliže se má certifikát protější strany komunikace kontrolovat certifikační autoritou (**verify_cert on**; tj. výchozí nastavení), potom se CA certifikát také musí uložit v tomto adresáři. Aby OpenSSL nalezlo tento certifikát, musí se přejmenovat pomocí hash jména nebo vytvořit symbolický odkaz (link):

```
ln -s CAfile.pem `openssl x509 -noout -hash < CAfile.pem`.0
```

Jestliže se má navíc kontrolovat certifikát proti souboru stornovaných certifikátů (CRL), musí se CRL uložit ve stejném adresáři a užívat hash jméno, vytvořené podobně jako výše pomocí symbolického odkazu:

```
ln -s CRLfile.pem `openssl x509 -noout -hash < CAfile.pem`.r0
```

Když ukládáte certifikáty a soukromý klíč, je důležité si všimnout, že **racoon** nemůže dešifrovat soukromý klíč. Proto se musí soukromý klíč uložit v nešifrovaném formátu čistého textu. Jestliže jste vytvořili šifrovaný soukromý klíč, musíte ho dešifrovat:

```
# openssl rsa -in my_private_key.pem -out my_private_key.pem
read RSA key
Enter PEM pass phrase: password
writing RSA key
```

Cestující uživatel

Cestující uživatelé (roadwarriors) jsou klienti, kteří užívají neznamé dynamické IP adresy pro připojení k VPN bráně. Ve spojení s démonem **racoon** vznikají dva problémy:

- IP adresa není známa a nemůže být specifikovaná v konfiguračním souboru démonu **racoon** nebo v souboru */etc/psk.txt*. Musí se nalézt jiný způsob určení identity klienta. Když se užívají sdílené klíče, vyžaduje to agresivní mód! Nejlepším řešením je ale použití X.509 certifikátů.
- Pro démon **racoon** nemůže být vytvořena žádná bezpečnostní politika, poněvadž cílová IP adresa není známa. **Racoon** musí vytvořit bezpečnostní politiku a SA v době, kdy se zahajuje spojení.

Aby se toho dosáhlo, konfigurační soubor */etc/racoon.conf* potřebuje několik změn:

```
path certificate "/etc/certs";

remote anonymous {
        exchange_mode main;
        generate_policy on;
        passive on;
        certificate_type x509 "my_certificate.pem" "my_private_key.pem";
        my_identifier asn1dn;
        peers_identifier asn1dn;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method rsasig;
                dh_group modp1024;
        }
}

sainfo anonymous {
        pfs_group modp1024;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

Direktiva **generate_policy on** nařizuje démonu **racoon** vytvořit příslušnou politiku v době, kdy se zahajuje spojení. Direktiva **passive on** říká démonu **racoon** zůstat pasivní a čekat až bude zvenku zahajované nové spojení. **Racoon** zahajovat spojení nesmí.

Nejdůležitější změna je ale definice **anonymous** v řádku **remote** a **sainfo**. Ta nařizuje démonu **racoon** přijmout spojení odkudkoliv.

NAT-Traversal

Linuxové jádro 2.6 má schopnost užívat NAT-Traversal v tunelovém módu. Toto může využívat **racoon** v ipsec-tools počínaje verzí 0.3.3. Transportní mód ještě není podporován.

Aby se mohl **racoon** konfigurovat pro NAT-Traversal, přidalo se do konfiguračního souboru několik direktiv. Jsou to **natt_keepalive**, **isakmp_natt**, **nat_traversal**.

Nejdůležitější direktiva je **nat_traversal**. Ta může být nastavena na **on**, **off** nebo **force**. Když je nastavena na **on,** účastník komunikace bude používat NAT-Traversal, jakmile je na cestě detekováno NAT zařízení. **Off** vypne toto chování. Když se užije **force**, NAT-Traversal se bude užívat bez ohledu na to, zda-li se NAT zařízení nalezne nebo ne.

Poněvadž mnoho NAT zařízení zapomene položky ve svých vnitřních tabulkách docela rychle, když není pozorován žádný datový provoz, démon **racoon** nabízí odesílání paketů udržujících existenci těchto položek (keepalive packets). Ve výchozím nastavení jsou odesílané každých 20 vteřin. Tuto hodnotu můžete měnit pomocí direktivy **natt_keepalive**. Nastavení direktivy na nulovou hodnotu vypne tuto vlastnost.

Jestliže chcete používat NAT-Traversal, musíte specifikovat užívanou IP adresu a port v sekci **listen** konfiguračního souboru démonu **racoon**. To se provádí pomocí direktivy **isakmp_natt**.

V zájmu jasného výkladu je prezentován typický konfigurační soubor, kde účastník komunikace 192.168.2.100 je skrytý za NAT bránou s IP adresou 192.168.1.1:

```
path pre_shared_key "/etc/psk.txt";

timer  {
       natt_keepalive 10sec;
       }

listen {
       isakmp 192.168.1.100 [500];
       isakmp_natt 192.168.1.100 [4500];
       }

remote 192.168.1.1 {
        exchange_mode main;
        nat_traversal on;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method pre_shared_key;
                dh_group modp1024;
        }
}
```

```
sainfo address 172.16.1.0/24 any address 172.16.2.0/24 any {
        pfs_group modp768;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

Jestliže jste vše konfigurovali správně, bude se NAT detekovat automaticky:

```
2004-12-22 07:34:53: INFO: @(#)ipsec-tools 0.4 (http://ipsec-
tools.sourceforge.net)
2004-12-22 07:34:53: INFO: @(#)This product linked OpenSSL 0.9.7a Feb 19 2003
(http://www.openssl.org/)
2004-12-22 07:34:53: INFO: 192.168.1.100[4500] used as isakmp port (fd=6)
2004-12-22 07:34:53: INFO: 192.168.1.100[4500] used for NAT-T
2004-12-22 07:34:53: INFO: 192.168.1.100[500] used as isakmp port (fd=7)
2004-12-22 07:35:09: INFO: respond new phase 1 negotiation:
192.168.1.100[500]<=>192.168.1.1[500]
2004-12-22 07:35:09: INFO: begin Identity Protection mode.
2004-12-22 07:35:09: INFO: received Vendor ID: draft-ietf-ipsec-nat-t-ike-02
2004-12-22 07:35:09: INFO: received Vendor ID: RFC XXXX
2004-12-22 07:35:09: INFO: Selected NAT-T version: RFC XXXX
2004-12-22 07:35:09: INFO: Hashing 192.168.1.100[500] with algo #1
2004-12-22 07:35:09: INFO: NAT-D payload #0 verified
2004-12-22 07:35:09: INFO: Hashing 192.168.1.1[500] with algo #1
2004-12-22 07:35:09: INFO: NAT-D payload #1 doesn't match
2004-12-22 07:35:09: INFO: NAT detected: PEER
2004-12-22 07:35:10: INFO: Hashing 192.168.1.1[500] with algo #1
2004-12-22 07:35:10: INFO: Hashing 192.168.1.100[500] with algo #1
2004-12-22 07:35:10: INFO: Adding remote and local NAT-D payloads.
2004-12-22 07:35:10: INFO: NAT-T: ports changed to: 192.168.1.1[4500]<-
>192.168.1.100[4500]
2004-12-22 07:35:10: INFO: KA list add: 192.168.1.100[4500]->192.168.1.1[4500]
2004-12-22 07:35:10: INFO: ISAKMP-SA established 192.168.1.100[4500]-
192.168.1.1[4500] spi:0613dc09c4ccc828:9cc9dfc9acc82eb5
2004-12-22 07:35:11: INFO: respond new phase 2 negotiation:
192.168.1.100[0]<=>192.168.1.1[0]
2004-12-22 07:35:11: INFO: Adjusting my encmode UDP-Tunnel->Tunnel
2004-12-22 07:35:11: INFO: Adjusting peer's encmode UDP-Tunnel(3)->Tunnel(1)
2004-12-22 07:35:11: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.1-
>192.168.1.100 spi=95762109(0x5b536bd)
2004-12-22 07:35:11: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.100-
>192.168.1.1 spi=222871470(0xd48bfae)
```

Když se díváte na pakety na lince (wire), budete vidět UDP provoz procházející tam a zpět:

```
[root@bibo root]# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap1, link-type EN10MB (Ethernet), capture size 96 bytes
13:37:41.920621 IP 192.168.1.1.isakmp > 192.168.1.100.isakmp: isakmp: phase 1 I
ident
13:37:41.941296 IP 192.168.1.100.isakmp > 192.168.1.1.isakmp: isakmp: phase 1 R
ident
13:37:42.051826 IP 192.168.1.1.isakmp > 192.168.1.100.isakmp: isakmp: phase 1 I
ident
13:37:42.157134 IP 192.168.1.100.isakmp > 192.168.1.1.isakmp: isakmp: phase 1 R
ident
13:37:42.353942 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 72
13:37:42.361530 IP 192.168.1.100.4500 > 192.168.1.1.4500: UDP, length 72
13:37:42.373799 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 88
```

```
13:37:43.374630 IP 192.168.1.100.4500 > 192.168.1.1.4500: UDP, length 1
13:37:43.384476 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 256
13:37:43.431219 IP 192.168.1.100.4500 > 192.168.1.1.4500: UDP, length 256
13:37:43.436680 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 56
13:37:44.492976 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 1
13:37:45.390137 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 116
13:37:45.390612 IP 192.168.1.100.4500 > 192.168.1.1.4500: UDP, length 116
13:37:46.395603 IP 192.168.1.1.4500 > 192.168.1.100.4500: UDP, length 116
13:37:46.396009 IP 192.168.1.100.4500 > 192.168.1.1.4500: UDP, length 116
```

Jestliže neužíváte démon **racoon** ve scénáři cestujícího uživatele, ale s pevnými adresami jako výše, potřebujete také upravit vaše bezpečnostní politiky. Ty pořebují zohledňovat "natované" adresy! Správné politiky pro scénář uvedený výše jsou:

```
#!/usr/sbin/setkey -f
#
# Flush SAD and SPD
flush;
spdflush;

# Create policies for racoon
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
           esp/tunnel/192.168.1.100-192.168.1.1/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
           esp/tunnel/192.168.1.1-192.168.1.100/require;
```

Tyto politiky se nastavují automaticky, jestliže ve vaší konfiguraci démonu **racoon** užíváte **generate_policy on.**

Linuxové jádro 2.6 užívající OpenBSD isakmpd

Thomas Walpuski přenesl démon IKE z operačního systému OpenBSD na Linux (http://bender.thinknerd.de/~thomas/IPsec/isakmpd-linux.html). Démon isakmpd se nyní může užívat na linuxovém jádře 2.5.47+ a 2.6.x, aby se nakonfigurovaly IPsec VPNs. Tato kapitola bude popisovat instalaci a konfiguraci démonu isakmpd.

Instalace

Jestliže užíváte distribuci založenou na typu RPM balíčků nebo Debian, může se instalace provést pomocí příslušných balíčkovacích nástrojů. Autor tohoto dokumentu kompiloval RPM balíček isakmpd démonu pro linuxové jádro 2.6.0 (http://www.spenneberg.org/VPN/Kernel-2_6_IPsec). Uvědomte si laskavě, že tento balíček možná nepracuje na testovacích verzích (jádra), protože se ABI v jádře několikrát změnilo. Projekt Debian zahrnuje balíček, který se může instalovat pomocí příkazu **apt-get install isakmpd**.

Když instalujete ze zdrojového kódu, potřebujete, jestliže chcete používat X.509 certifikáty, keynote balíček (http://www1.cs.columbia.edu/~angelos/keynote.html). Kromě toho potřebujete linuxové jádro 2.5.47+ nebo 2.6.x.

Aby jste obdrželi zdrojové kódy démonu isakmpd, sledujte kroky zmíněné na webové stránce Thomase Walpuského. Potom podle toho editujte GNU Makefile a aktivujte řádek **OS=linux**. Jestliže

nemáte linuxové jádro v adresáři */usr/src/linux*, budete navíc potřebovat upravit soubor *sysdep/linux/GNUmakefile.sysdep*.

Kompilace se může provést pomocí příkazu **make**.

Démon isakmpd doprovázejí další dva příkazy: **keyconv** a **certpatch**. Tyto nástroje jsou v podadresáři *apps* a mohou se kompilovat ručně. (Jsou součástí mého RPM balíčku.) **Certpatch** může přidat SubjectAltName do existujícího certifikátu, zatímco **keyconv** převádí (converts) DNSSEC na openssl klíče a naopak.

Tyto nástroje jsem úspěšně mohl kompilovat pomocí (Vaše kompilační volby se ale mohou lišit.):

```
gcc -DMP_FLAVOUR=MP_FLAVOUR_GMP -I../.. -I../../sysdep/linux   -I /usr/src/linux-
2.6.0 -lcrypto -lgmp certpatch.c -o certpatch
gcc  -I../.. -I../../sysdep/linux   -I /usr/src/linux-2.6.0 -lcrypto -lgmp
base64.c keyconv.c -o keyconv
```

Poslední varování: Veškeré stránky dokumentace man jsou v kódové stránce Latin1. Red Hat 9 neumí zobrazit tyto stránky. Musíte je převést, aby jste je mohli číst (, což je v RPM balíčku již uděláno):
iconv --from-code LATIN1 --to-code UTF-8 --output isakmpd2.8 isakmpd.8

Když už je isakmpd zkompilován, vytvořte povinnou adresářovou strukturu:

```
mkdir /etc/isakmpd
mkdir /etc/isakmpd/ca
mkdir /etc/isakmpd/certs
mkdir /etc/isakmpd/keynote
mkdir /etc/isakmpd/crls
mkdir /etc/isakmpd/private
mkdir /etc/isakmpd/pubkeys
```

Použití sdílených klíčů (PSK)

Démon isakmpd užívá jeden konfigurační soubor a jeden soubor pro politiku. Jsou to */etc/isakmpd/isakmpd.conf* a */etc/isakmpd/isakpmd.policy*. Styl konfigurace se opírá o dobře známý .INI formát. Každá sekce začíná řádkem podobně jako:

```
[section]
```

V rámci sekce můžete proměnné přiřadit někajou hodnotu:

```
tag=value
```

Jestliže je hodnota delší než jeden řádek, můžete použít techniku zpětného lomítka (backslash), aby jste ji rozdělili na několik řádek. Komentáře mohou být kladeny kdekoliv pomocí znaku # (hash mark).

Abychom začali, podíváme se na jednoduchou konfiguraci, která užívá pro ověření identity sdílený klíč (pre-shared secret). Pro situaci (setup), kterou budeme řešit, se laskavě podívejte se na obrázek 5 v odstavci Tunelový mód.

```
[General]
Listen-on=              192.168.1.100
```

```
[Phase 1]
192.168.2.100=          ISAKMP-peer-west

[Phase 2]
Connections=            IPsec-east-west

[ISAKMP-peer-west]
Phase=                  1
Local-address=          192.168.1.100
Address=                192.168.2.100
Authentication=         ThisIsThePassphrase

[IPsec-east-west]
Phase=                  2
ISAKMP-peer=            ISAKMP-peer-west
Configuration=          Default-quick-mode
Local-ID=               Net-east
Remote-ID=              Net-west

[Net-west]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.2.0
Netmask=                255.255.255.0

[Net-east]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.1.0
Netmask=                255.255.255.0

[Default-quick-mode]
DOI=                    IPSEC
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-MD5-PFS-SUITE
```

Konfigurační soubor popisuje tunel mezi dvěma branami s IP adresami 192.168.1.100 a 192.168.2.100. Tunel mohou využívat (pod)sítě s adresami 172.16.1.0/24 a 172.16.2.0/24. Tento konfigurační soubor je speciálně určen pro bránu 192.168.1.100.

Podívejme se na jednotlivé sekce. První sekce [General] popisuje všeobecné nastavení. Zde definujeme, jestli by se démon isakmpd měl během startu vázat (bind) ke specifickým IP adresám. To se doporučuje, jestliže máte na vaší VPN bráně několik IP adres.

Sekce [Phase 1] popisuje, kterou konfiguraci použít pro protější stranu komunikace s IP adresou 192.168.2.100. Jestliže IP adresa protější strany komunikace není známa (cestující uživatel), můžete místo ní použít default.

Sekce [Phase 2] popisuje tunely, které se mají vytvořit, jakmile se prokázala identita, Phase 1. Jestliže démon isakmpd nesmí aktivně zahajovat spojení, použijte místo toho **Passive-connections**.

Nyní musíte definovat jména, ke kterým jste se odkazovali v sekcích Phase 1 a Phase 2. Nejdříve definujeme ISAKMP-peer-west. Tato definice se užívá v sekci **Phase 1** a známe místní adresu **Local-address** a vzdálenou adresu **Address**. Jestliže vzdálená adresa není známa, pouze smažte tuto proměnou. Ověření identity **Authentication** by mělo být provedeno pomocí sdíleného klíče, který je zadán ve formě čistého textu.

Dále se definuje tunel IPsec-east-west. Je užívaný v sekci **Phase 2** a bude zřízen s proměnou **ISAKMP-peer** nastavenou na hodnotu ISAKMP-peer-west. Chceme definovat konfiguraci spojení **Configuration** a další identifikátory pro tunel (**Local-ID** a **Remote-ID**). Poněvadž tyto identifikátory jsou opět odkazy, musíme je definovat. Proměnná **ID-type** může nabývat hodnot IPV4_ADDR, IPV6_ADDR, IPV4_ADDR_SUBNET a IPV6_ADDR_SUBNET.

V neposlední řadě se musí definovat konfigurace rychlého módu (quick-mode), ke které jsme se odkazovali v popisu tunelu. Definujeme proměnné **DOI** (výchozí hodnota: IPSEC), **EXCHANGE_TYPE** (výchozí hodnota: QUICK_MODE) a **Suites**. Tj. QuickMode-Encapsulated-Security-Payload-3DES-Encryption-MD5-HMAC-Perfect-Forward-Secrecy. Pro proměnou Suites lze definovat několik hodnot oddělených čárkou. Pro možné hodnoty si přečtěte stránky dokumentace man.

Soubor *isakmpd.policy* je mnohem kratší. Následující výpis ukazuje příklad:

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: "passphrase:ThisIsThePassphrase"
Conditions: app_domain == "IPsec policy" &&
            esp_present == "yes" &&
            esp_enc_alg == "3des" &&
            esp_auth_alg == "hmac-md5" -> "true";
```

Aby jste spojení otestovali, spustěte démon isakmpd pomocí následujícího řádku:

```
isakmpd -d -4 -DA=90
```

To bude startovat isakmpd v popředí (-d) s použitím IPv4 (-4) a úrovní ladění (debuglevel) 90.

Jakmile spojení nastartovalo, měli by jste být schopni provést ping z jedné podsítě na druhou. Jestliže jste také nainstalovali balíček ipsec-tools, můžete užít příkaz **setkey**, aby jste viděli politiky a SAs přidané démonem isakmpd. Jestliže ukončíte práci démonu isakmpd běžícího v popředí pomocí CTRL-C, démon nevyčistí databáze SAD a SPD. Musíte to provést ručně pomocí příkazu **setkey**. Jestliže ukončíte isakmpd pomocí příkazu **kill -TERM**, vyčistí databáze SAD a SPD!

Použití X.509 certifikátů

Démon isakmpd také může pro proces ověření identity užívat X.509 certifikáty. Vaše certifikáty můžete vytvořit pomocí obvyklých nástrojů a pro každý počítač zůčastěný ve VPN potřebujete následující soubory:

- */etc/isakmpd/private/local.key*. Soukromý klíč počítače ve formátu .pem. Přístupová práva (permissions) musejí být 600.
- */etc/isakmpd/ca/ca.crt*. Certifikát certifikační autority, které důvěřujete.
- */etc/isakmpd/certs/ip-address.crt*. Certifikát místního počítače.

Certifikát musí zahrnovat SubjectAltName, aby jej démon isakmpd nalezl a užíval. Toto rozšíření X509v3 může být definované během generování certifikátu nebo později pomocí příkazu **certpatch**. Tento příkaz potřebuje soukromý klíč certifikační autority CA, vytáhne (extracts) certifikát, přidává rozšíření a certifikát zase podepíše.

```
certpatch -i ip-address -k ca.key  originalcert.crt newcert.crt
```

Certpatch může do certifikátu přidávat IP adresu, FQDN nebo UFQDN.

Jakmile jsou tyto soubory uložené do příslušných adresářů a mají nastavené příslušné přístupové práva, můžete vytvořit konfigurační soubor a soubor pro politiku. V konfiguračním souboru v sekci ISAKMP-peer-west pouze smažte řádek Authentication a přidejte řádek **ID=East**. Potom definujte East. Kromě toho musíte specifikovat X.509 adresáře. Následuje úplný konfigurační soubor:

```
[General]
Listen-on=              192.168.1.100

[Phase 1]
192.168.2.100=          ISAKMP-peer-west

[Phase 2]
Connections=            IPsec-east-west

[ISAKMP-peer-west]
Phase=                  1
Local-address=          192.168.1.100
Address=                192.168.2.100
ID=                     East

[East]
ID-type=                IPV4_ADDR
Address=                192.168.1.100

[IPsec-east-west]
Phase=                  2
ISAKMP-peer=            ISAKMP-peer-west
Configuration=          Default-quick-mode
Local-ID=               Net-east
Remote-ID=              Net-west

[Net-west]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.2.0
Netmask=                255.255.255.0

[Net-east]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.1.0
Netmask=                255.255.255.0

[Default-quick-mode]
DOI=                    IPSEC
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-MD5-PFS-SUITE

[X509-certificates]
CA-directory=           /etc/isakmpd/ca/
Cert-directory=         /etc/isakmpd/certs/
Private-key=            /etc/isakmpd/private/local.key
```

Také se musí modifikovat soubor pro politiku. Poněvadž chcete povolit použití pouze pro účastníky komunikace, kteří disponují certifikáty podepsanými důvěryhodnou autoritou CA, přidejte za řádek Authorizer následující řádek. Následuje úplný soubor pro politiku:

```
KeyNote-Version: 2
```

```
Authorizer: "POLICY"
Licensees: "DN:/C=DE/ST=NRW/L=Steinfurt/O=Spenneberg.Com/OU=VPN/CN=RootCA"
Conditions: app_domain == "IPsec policy" &&
            esp_present == "yes" &&
            esp_enc_alg == "3des" &&
            esp_auth_alg == "hmac-md5" -> "true";
```

Text po **DN:** se musí rovnat řádku subject v CA certifikátu:

```
openssl x509 -in ca/ca.crt -noout -subject
```

Nyní můžete obvyklým způsobem spustit démon isakmpd, aby jste konfiguraci otestovali.

Generování X.509 certifikátů

Téměř všechny VPN implementace dnes umožňují pro ověřování identity účastníků komunikace použití X.509 certifikátu. Jsou to stejné certifikáty, které se užívají pro implementaci SSL (Secure Socket Layer) v HTTP protokolu.

Tato kapitola se bude stručně zabývat tvorbou těchto certifikátů.

Užití OpenSSL

Nejsnadnější cestou pro vytvoření X.509 certifikátů na Linuxu je příkaz **openssl** a pomocné nástroje. Když byl nainstalovaný balíček openssl, je obvykle také instalován pomocný příkaz **CA** a/nebo **CA.pl**. Pro tvorbu certifikátů budeme užívat tento příkaz.

Nejdříve prověřte, kde je příkaz nainstalovaný. Obvykle není ve vaší systémové cestě! Na distribucích Red Hat Linux je uložen v */usr/share/ssl/misc/CA*.

Nyní nejdříve vytvořte certifikační autoritu.

```
$ mkdir certs
$ cd certs
$ /usr/share/ssl/misc/CA -newca
CA certificate filename (or enter to create) <enter>

Making CA certificate ...
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
................++++++
.............++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase: capassword
Verifying password - Enter PEM pass phrase: capassword
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
```

```
State or Province Name (full name) [NRW]:
Locality Name (eg, city) [Steinfurt]:
Organization Name (eg, company) [Spenneberg.com]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:RootCA 2003
Email Address []:ralf@spenneberg.net
```

Když budete žádáni o Country Name atd., laskavě zadejte příslušné hodnoty. Jestliže chcete mít nabízené správné hodnoty (podobně jako v případě výše), editujte váš soubor *openssl.cnf*. Na systémech Red Hat Linux jej obvykle můžete nalézt v adresáři */usr/share/ssl*.

Vytvořená certifikační autorita je platná pouze jeden rok. Často chcete delší životnost pro váš CA certifikát. Poněvadž certifikáty, které budete podepisovat později, obvykle mají kratší životnost, není praktické editovat soubor *openssl.cnf*. Životnost spíše změňte ručně:

```
$ cd demoCA/
$ openssl x509 -in cacert.pem -days 3650 -out cacert.pem
-signkey ./private/cakey.pem
Getting Private key
Enter PEM pass phrase: capassword
$ cd ..
```

Certifikační autorita je nyní připravena fungovat. Vytvořme žádost na podepsání certifikátu (Certificate Signing Request):

```
$ /usr/share/ssl/misc/CA -newreq
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
..............................++++++
.................................++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase: certpassword
Verifying password - Enter PEM pass phrase: certpassword
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [NRW]:
Locality Name (eg, city) [Steinfurt]:
Organization Name (eg, company) [Spenneberg.com]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:VPN-Gateway
Email Address []:ralf@spenneberg.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
```

Soubor *newreq.pem* obsahuje žádost na podepsání certifikátu a zašifrovaný soukromý klíč. Tento soubor se může později použít jako soukromý klíč pro FreeS/WAN nebo démon racoon. Jakmile je žádost vytvořena, můžeme ji podepsat pomocí certifikační autority.

```
$ /usr/share/ssl/misc/CA -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: capassword
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'NRW'
localityName          :PRINTABLE:'Steinfurt'
organizationName      :PRINTABLE:'Spenneberg.com'
commonName            :PRINTABLE:'VPN-Gateway'
emailAddress          :IA5STRING:'ralf@spenneberg.net'
Certificate is to be certified until Apr 29 06:08:56 2004 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

V závislosti na verzi příkazu **CA** se může certifikát vytisknout na standardní výstup (stdout). Bude to podobné následujícímu certifikátu:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=DE, ST=NRW, L=Steinfurt, O=Spenneberg.com,
CN=RootCA 2003/Email=ralf@spenneberg.net
        Validity
            Not Before: Apr 30 06:08:56 2003 GMT
            Not After : Apr 29 06:08:56 2004 GMT
        Subject: C=DE, ST=NRW, L=Steinfurt, O=Spenneberg.com,
CN=VPN-Gateway/Email=ralf@spenneberg.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c5:3b:9c:36:3a:19:6c:a9:f2:ba:e9:d2:ed:84:
                    33:36:48:07:b2:a3:2d:59:92:b0:86:4c:81:2c:ea:
                    5c:ed:f3:ba:eb:17:4e:b3:3a:cc:b7:5b:5d:ca:b3:
                    04:ed:fb:59:3c:c5:25:3e:f3:ff:b0:22:10:fb:de:
                    72:0a:ee:42:4b:9a:d3:27:d3:b6:fb:e9:88:10:c8:
                    47:b7:26:4f:71:40:e4:75:c4:c0:ee:6b:87:b8:6f:
                    c9:5e:66:cf:bb:e7:ad:72:68:b8:6d:fd:8f:4c:1f:
                    3a:a2:0d:43:25:06:b9:92:e7:20:6c:86:15:a0:eb:
                    7f:f7:0b:9a:99:5d:14:88:9b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                CB:5C:19:9B:E6:8A:8A:FE:0E:C4:FD:5E:DF:F7:BF:3D:A8:
18:7C:08
            X509v3 Authority Key Identifier:
                keyid:01:BB:C6:33:BE:F5:9A:5E:B0:0C:5D:BD:41:E9:78:
6C:54:AD:66:8E
```

```
                DirName:/C=DE/ST=NRW/L=Steinfurt/O=Spenneberg.com/
CN=RootCA 2003/Email=ralf@spenneberg.net
                serial:00

    Signature Algorithm: md5WithRSAEncryption
        6f:89:2b:95:af:f1:8d:4d:b7:df:e8:6d:f7:92:fb:48:8c:c4:
        1a:43:68:65:97:01:87:a6:84:b5:a1:38:bd:62:74:70:db:9e:
        78:19:d9:0c:af:18:ad:13:77:56:7d:3f:19:61:da:ba:74:30:
        8e:c5:50:0e:e3:eb:ff:95:cd:8d:d6:7e:c3:0e:ab:5b:34:94:
        bc:16:0f:ef:dc:de:40:bb:7d:ba:a2:b8:5d:f9:74:e7:28:58:
        75:a0:66:d2:8d:85:ba:38:82:08:10:33:ef:be:29:c9:31:9d:
        63:a9:f7:e0:99:ea:a7:ed:b6:b5:33:1b:1c:4a:a4:05:40:6e:
        40:7b
-----BEGIN CERTIFICATE-----
MIIDjDCCAvWgAwIBAgIBATANBgkqhkiG9w0BAQQFADCBgjELMAkGA1UEBhMCREUx
DDAKBgNVBAgTA05SVzESMBAGA1UEBxMJU3RlaW5mdXJ0MRcwFQYDVQQKEw5TcGVu
bmVpZXJnLmNvbTEUMBIGA1UEAxMLUm9vdENBIDIwMDMxIjAgBgkqhkiG9w0BCQEW
E3JhbGZAc3Blbm5lYmVyZy5uZXQwHhcNMDMwNDMwMDYwODU2WhcNMDQwNDI5MDYw
ODU2WjCBgjELMAkGA1UEBhMCREUxDDAKBgNVBAgTA05SVzESMBAGA1UEBxMJU3Rl
aW5mdXJ0MRcwFQYDVQQKEw5TcGVubmViZXJnLmNvbTEUMBIGA1UEAxMLVlBOLUdh
dGV3YXkxIjAgBgkqhkiG9w0BCQEWE3JhbGZAc3Blbm5lYmVyZy5uZXQwgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAMU7nDY6GWyp8rrp0u2EMzZIB7KjLVmSsIZM
gSzqXO3zuusXTrM6zLdbXcqzBO37WTzFJT7z/7AiEPvecgruQkua0yfTtvvpiBDI
R7cmT3FA5HXEwO5rh7hvyV5mz7vnrXJouG39j0wfOqINQyUGuZLnIGyGFaDrf/cL
mpldFIibAgMBAAGjggEOMIIBCjAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1P
cGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUy1wZm+aKiv4O
xP1e3/e/PagYfAgwga8GA1UdIwSBpzCBpIAUAbvGM771ml6wDF29Qel4bFStZo6h
gYikgYUwgYIxCzAJBgNVBAYTAkRFMQwwCgYDVQQIEwNOUlcxEjAQBgNVBAcTCVN0
ZWluZnVydDEXMBUGA1UEChMOU3Blbm5lYmVyZy5jb20xFDASBgNVBAMTC1Jvb3RD
QSAyMDAzMSIwIAYJKoZIhvcNAQkBFhNyYWxmQHNwZW5uZWJlcmcubmV0ggEAMA0G
CSqGSIb3DQEBBAUAA4GBAG+JK5Wv8Y1Nt9/obfeS+0iMxBpDaGWXAYemhLWhOL1i
dHDbnngZ2QyvGK0Td1Z9Pxlh2rp0MI7FUA7j6/+VzY3WfsMOq1s0lLwWD+/c3kC7
fbqiuF35dOcoWHWgZtKNhbo4gggQM+++KckxnWOp9+CZ6qfttrUzGxxKpAVAbkB7
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
```

Nyní je rozumné přejmenovat soubory *newreq.pem* a *newcert.pem* k něčemu více smysluplnému.

```
$ mv newcert.pem vpngateway_cert.pem
$ mv newreq.pem vpngateway_key.pem
```

Nyní se můžete pobavit tvorbou certifikátů pro každého účastníka VPN komunikace.

V případě, že je soukromý klíč ukraden nebo je ohrožena jeho bezpečnost, musíte ho zrušit (revoke), protože v závislosti na jeho životnosti může ještě stále být platný. Zrušené klíče jsou uložené v seznamu stornovaných certifikátů (Certificate Revocation List, CRL). Nejdříve vytvořte (prázdný) seznam:

```
$ openssl ca -gencrl -out crl.pem
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: capassword
```

Aby jste zrušili certifikát, musíte mít soubor certifikátu. Ten je také uložen v *demoCA/newcerts/*. Jméno certifikátu se může přečíst v *demoCA/index.txt*. Potom použijte následující příkaz.

```
$ openssl ca -revoke compromised_cert.pem
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: capassword
```

```
Revoking Certificate 01.
Data Base Updated
```

Jakmile byl certifikát zrušen, musí se seznam stornovaných certifikátů znovu vytvořit pomocí výše uvedeného příkazu.

Generování certifikátů pro Windows klienty

Když tvoříte certifikáty pro Windows klienty, musíte zajistit, aby životnost certifikátu ležela v rámci času životnosti certifikační autority CA. Jestliže životnost certifikátu přesahuje životnost CA, Windows klient nepřijme certifikát!

Nejsnadnější způsob, jak přenést certifikáty na Windows počítač, je pomocí PKCS#12 formátu. Openssl může přeformátovat certifikáty na tento formát:

```
$ openssl pkcs12 -export -inkey key.pem -in cert.pem -certfile cacert.pem -out
export.p12 -name "Windows Cert"
```

Jste žádáni specifikovat exportní heslo. Na Windows počítači potom můžete importovat tento soubor pomocí tohoto exportního hesla.

Nástroj, který by vám snad mohl pomoci při generování souboru ve formátu PKCS#12, je Wincert. URL k tomuto nástroji naleznete v sekci odkazů (links).

Pokročilá konfigurace

Xauth a IKE-Mde-Config

Bohužel Xauth a IKE-Mode-Config jsou nefunkční na Linuxu, jež užívá ipsec-tools <= 0.6. Jakmile bude Xauth fungovat, uvedu nějaké pokyny pro jeho používání.

IPtables pravidla

ESP v tunelovém módu bez komprese zvětšuje velikost přenášených paketů. To se někdy dokonce i stane, když je komprese aktivovaná. V tunelovém módu to může způsobit problémy, když klient neví, že paket je zapouzdřený. Jestliže klienti posílají paket o velikosti 1500 bajtů, dodatečné zapouzdření zvětší velikost paketu. Pro TCP pakety můžete tento problém řešit nastavením MSS na obou stranách tunelu pomocí příkazu **iptables**:

```
iptables -t mangle -A PREROUTING -p esp -j MARK --set-mark 1
iptables -A FORWARD -m mark --mark 1 -p tcp --tcp-flags SYN,RST SYN -j
TCPMSS --set-mss 1400
```

Odkazy

Tato sekce poskytuje nějaké odkazy na nástroje, které by jste snad mohli potřebovat.

- IPsec-Tools: http://ipsec-tools.sf.net

- ipsec.exe Markuse Muellera pro připojení Windows počítačů k VPN: http://vpn.ebootis.de
- Wincert pomáhá v generování certifikátů ve formátu PKCS#12: http://sourceforge.net/projects/wincert/

Poznámky

1. http://www.tldp.org
2. http://www.ipsec-howto.org
3. http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html
4. http://www.tldp.org/HOWTO/Net-HOWTO/index.html
5. http://www.tldp.org/HOWTO/VPN-Masquerade-HOWTO.html
6. http://www.tldp.org/HOWTO/VPN-HOWTO/
7. http://www.tldp.org/HOWTO/Adv-Routing-HOWTO/
8. http://www.kernel.org
9. http://ipsec-tools.sourceforge.net
10. http://bender.thinknerd.de/~thomas/IPsec/isakmpd-linux.html
11. http://www.spenneberg.org/VPN/Kernel-2_6_IPsec
12. http://www1.cs.columbia.edu/~angelos/keynote.html
13. http://ipsec-tools.sf.net
14. http://vpn.ebootis.de
15. http://sourceforge.net/projects/wincert/