

Tworzenie szyfrowanych tuneli z wykorzystaniem implementacji IPsec z kerneli 2.6.x

VPN - nowe rozdanie.

W sierpniowym wydaniu LinuxMagazine opisałem o tym jak w prosty sposób za pomocą protokołu PPTP stworzyć szyfrowany tunel pomiędzy LANem a komputerem pracującym pod dowolnym systemem operacyjnym wyposażonym w program-klienta PPTP. Ponieważ w wielu publikacjach wykazano, że PPTP nie jest najlepszym możliwym protokołem, który mógłby być wykorzystywany w celu zabezpieczenia naszych danych, należy poszukać jakiejś alternatywy. IPSec (IP Security), któremu poświęcony jest niniejszy tekst jest protokołem znacznie odmiennym w stosunku do PPTP, zarówno pod kontem samego bezpieczeństwa ogólnie rozumianego, jak i skomplikowania w budowie i użyciu.

IP Security od podstaw.

W ostatnich latach XX wieku protokół IP stał się dominującym sposobem na komunikację w Internecie. Wyparł zupełnie wszelkie inne pomysły na komunikację i stał się obowiązującym standardem. Dzięki niemu problem z komunikacją w obrębie globalnej sieci został dość dobrze rozwiązany, natomiast bardzo szybko pojawił się problem bezpieczeństwa przesyłanych informacji. Tu pojawia się nasz główny bohater – IPSec.

IPSec powstał ok. roku 1992 przy znacznym wsparciu IETF (Internet Engineering Task Force) ze względu na poważne niedoskonałości protokołu IP, który już wtedy zyskał dominującą pozycję w Internecie. Struktura sieci Internet i budowa IP zapewniała możliwość skomunikowania się pomiędzy dwoma odległymi sieciami mimo uszkodzeń wewnątrz poszczególnych elementów Internetu pomiędzy nimi. IP nie zajmuje się zupełnie (no prawie;)) sprawą poufności, integralności danych czy sprawdzania autentyczności rozmówców, co stworzyło zapotrzebowanie na protokół IPSec.

Ponieważ powstawanie bezpiecznego protokołu zbiegło się z koniecznością zajęcia się tematem ograniczonej ilości adresów IPv4 przy niespodziewanym wzroście Internetu, IPSec powstawał praktycznie równoległe z protokołem IPv6, którego stał się integralną częścią. Mimo tego IPSec został tak zaprojektowany aby można go było bez trudu wykorzystać także w sieciach opartych na IPv4, przewidując dość długi czas potrzebny na wdrożenie IPv6.

Trzy główne funkcje jakie miał zapewnić i zapewnia IPSec:

- *integralność*, czyli zapewnienie, że żadne dane wymieniane pomiędzy rozmówcami nie zostały zmienione
- *poufność*, czyli zapewnienie zabezpieczenia danych wymienianych pomiędzy rozmówcami przed możliwością podsłuchania
- *autentyczność* rozmówców, czyli pewność, że rozmówcy są naprawdę tymi za których się podają

Te trzy podstawowe elementy są zapewniane przez protokoły wchodzące w skład IPSec: AH (Authentication Header), ESP (Encapsulated Security Payload), oraz IKE (Internet Key Exchange).

Authentication Header

Wykorzystując kryptograficzne funkcje skrótu z dodatkowym użyciem klucza HMAC¹ (Hash Message Authentication Code) AH pozwalają na ochronę integralności pakietu IP, czyli jego nienaruszone doręczenie. Do obliczenia HMAC na podstawie klucza oraz zawartości pakietu IP wykorzystywane są np. algorytmy takie jak MD-5 czy SHA-1. Skrót ten jest następnie umieszczany w nagłówku pakietu IPSec, dzięki czemu odbiorca mający dostęp do klucza może sprawdzić czy dany pakiet, który do niego dotarł, nie został w trakcie transportu zmieniony i czy pochodzi od zaufanego nadawcy. AH przynosi w nagłówku pakietu IP także kolejny numer sekwencyjny połączenia zapobiegając włączeniu się nieautoryzowanych rozmówców. IANA przypisała numer 51 protokołowi AH, o czym trzeba pamiętać przy konfiguracji firewalla.

Encapsulated Security Payload

ESP zajmuje się zapewnieniem poufności przekazywanej informacji. Co ciekawe, podobnie jak AH, zapewnia także ochronę integralności danych obejmującą jedynie dane znajdujące się poza nagłówkiem IP. Integralność zapewniana jest za pomocą podobnych metod jak w przypadku AH, natomiast do zapewnienia poufności wykorzystywane są algorytmy, takie jak np. DES czy bardziej bezpieczne jak 3DES, AES lub Blowfish. W zależności od trybu pracy ESP może szyfrować jedynie część warstwy transportowej (np. TCP, UDP, ICMP, IGMP) lub cały pakiet. Nagłówek ESP jest wtedy umieszczany za nagłówkiem IP. IANA przypisała numer 50 protokołowi ESP, o czym trzeba pamiętać konfigurując firewall.

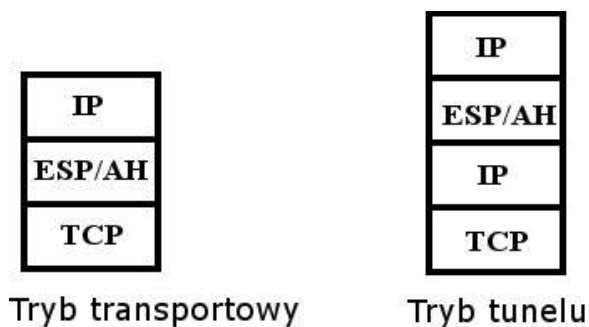
Internet Key Exchange

Ostatnim z głównych elementów IPSec jest IKE. Składa się z dwóch składowych: ISAKMP (Internet Security Association and Key Management Protocol), którego zadaniem jest negocjacja parametrów połączenia IPSec, oraz Oakley/SKEME zajmujący się wymianą kluczy za pomocą algorytmu Diffie-Hellmana. W pierwszej fazie IKE wymienione zostają ustalone klucze za pomocą algorytmu DH, aby ustanowić połączenie szyfrowane, a następnie - w fazie drugiej - ustanowione zostają SA czyli docelowe połączenia. IKE zajmuje się także zmianą okresową kluczy używanych do szyfrowania danych.

Ostatecznie mamy zestaw trzech standardów, za pomocą których ustanawiane jest szyfrowane połączenie IPSec.

Dodatkowo, w zależności od potrzeb, IPSec może pracować w trybie transportującym (transport mode) oraz tunelującym (tunnel mode). Pierwszy polega na ochronie jedynie części z danymi, umieszczając dodatkowy nagłówek IPSec pomiędzy nagłówkiem IP i nagłówkiem protokołu wyższej warstwy. Tryb tunelu szyfruje całość przesyłanych danych, nagłówek IP, oraz pozostałe dane, które zostają opakowane nowo utworzonym nagłówkiem IP za pomocą IPSec. Dzięki temu aplikacje warstw najwyższych modelu OSI/ISO nie są zupełnie świadome istnienia połączenia IPSec pomiędzy komunikującymi się hostami. Tryb transportowy wykorzystywany jest najczęściej pomiędzy dwoma hostami, natomiast tryb tunelu - pomiędzy dwoma bramkami IPSec mającymi za zadanie połączenie ze sobą dwóch lub więcej sieci IP.

1 Więcej o HMAC pod adresem: <http://www.ipsec.pl/leksykon/hmac.php>



Rys. 1 IPsec udostępnia dwa możliwe tryby pracy: transportowy i tunelu.

SA, SPI, SAD, SPD

Skoro znamy już podstawowe elementy IPsec, należałoby teraz poznać jak są one wykorzystywane przez system operacyjny i używaną przez nas implementację IPsec.

Każdy ze skonfigurowanych przez nas tuneli opisywany jest jako SA (Security Association). Jest to zestaw parametrów opisujących dany tunel. Każde SA opisuje elementy takie jak: źródłowy i docelowy adres IP bramki IPsec, protokoły wykorzystywane przez AH i ESP, czas ważności sesji, typ połączenia (transport, tunel) i kilka innych parametrów. Żeby było bardziej ciekawie, SA opisuje jedynie zestaw parametrów w jednym kierunku. Jeżeli konfigurujemy połączenie IPsec pomiędzy hostami A i B, potrzebne będzie określenie dwóch SA, po jednym w każdym kierunku, z A do B i z B do A. Pozwala to na różnicowanie algorytmów wykorzystywanych w danym SA w zależności od potrzeb i pozostałych parametrów, które są elementami SA.

Każde SA identyfikowane jest przez 32-bitową liczbę SPI (Security Parameters Index). Jest to jedyna informacja o szyfrowanym połączeniu, jaka może być podsłuchana w sieci, aczkolwiek SPI ma określone znaczenie jedynie dla uczestników danego tunelu, więc w praktyce jej podsłuchanie jest niewiele warte.

Wszystkie informacje o skonfigurowanych połączeniach SA przechowywane są w bazach SAD (SA Database). Decyzja o tym co należy począć z pakietem trafiającym do kernela systemu podejmowana jest natomiast na podstawie SP (Security Policy), która jest oczywiście przechowywana w bazie SPD (Security Policy Database).

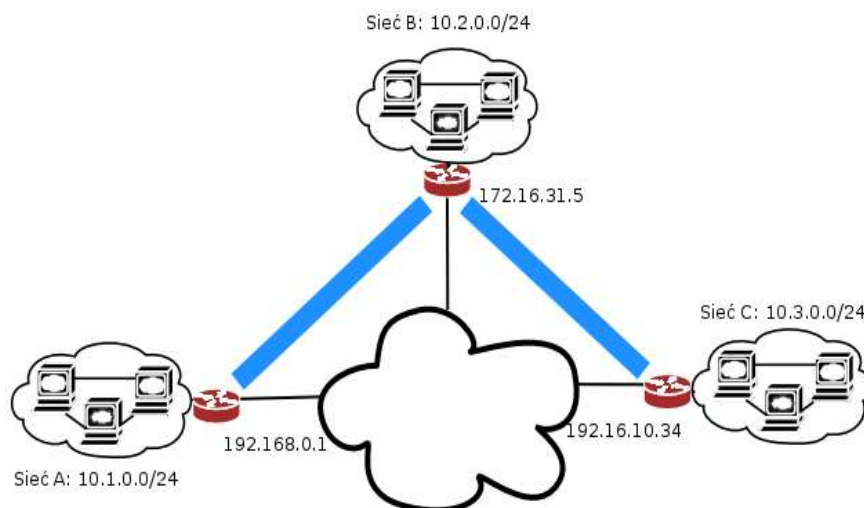
Reasumując, pakiet trafający od użytkownika w bramce IPsec poddawany jest sprawdzeniu w SPD czy pasuje do którejś z SP i czy powinien być skierowany do któregoś z ustanowionych połączeń IPsec. Jeśli tak, wykorzystywane jest wskazane połączenie, które zostało zestawione z bazy SAD na podstawie wcześniej opisanych SA. W skrócie: SP opisuje to co szyfrujemy, a SA – jak.

Przykładowa konfiguracja.

Ponieważ mamy już pokrótce opisaną strukturę połączeń IPsec i wyjaśnione podstawowe pojęcia, warto przejść do praktyki. Konfiguracja przedstawiona poniżej będzie bazowała na

implementacji IPSec, która jest dostępna w kernelach 2.6. Jest ona oparta na implementacji pochodzącej z projektu <http://www.kame.net/>, którego celem było stworzenie wsparcia dla IPv6 i IPSec w systemach BSD.

Zanim przystąpimy do właściwej konfiguracji, poniżej rysunek interesującego nas schematu sieci i planowanych tuneli.



Rys. 2 Schemat sieci, która została wykorzystana w przykładowej konfiguracji.

Mamy tu trzy sieci A, B i C, podłączone do internetu za pomocą bramek, które są połączone z Internetem za pomocą adresów 192.168.0.1, 172,16,31,5 i 192,16,10,34 odpowiednio. Są to ich adresy widziane z Internetu. Niebieskie linie oznaczają tunele IPSec, które chcemy stworzyć. Sieć A ma mieć zapewniony dostęp do sieci B, i podobnie sieć C do sieci B. Na razie nie ma mowy o połączeniu sieci A z C.

Na początek należy przygotować kernel. Na szczęście, czasy projektu FreeS/WAN² mamy już za sobą i IPSec jest dostępny w nowszych kernelach z serii 2.6. Wystarczy wybrać następujące opcje podczas kompilacji:

```
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
```

2 Projekt FreeS/WAN (<http://www.freeswan.org/>) był jedną z najbardziej popularnych implementacji IPSec do czasu ukazania się supportu w kernelach 2.6. Obecnie projekt ten nie jest już rozwijany, a jego spadkobiercą stał się projekt Openswan (<http://www.openswan.org/>).

```
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
...
```

W sekcji "Cryptographic API" należy wybrać protokoły, z których będziemy chcieli korzystać.

Po przygotowaniu nowego jądra trzeba dodatkowo zainstalować dwie rzeczy: pakiety dostarczające potrzebne oprogramowanie w Debianie dostępne są w paczkach *racoon* i *ipsec-tools*.

```
# apt-get install racoon
# apt-get install ipsec-tools
```

W przypadku innych systemów należy poszukać odpowiadających paczek lub skompilować obydwie rzeczy bezpośrednio ze źródeł, które dostępne są po adresem: <http://ipsec-tools.sourceforge.net/>

Pakiet *racoon* zawiera demona o tej samej nazwie, a *ipsec-tools* program *setkey*. Utworzony też zostaje katalog `/etc/racoon`, w którym umieścimy całą konfigurację potrzebną do stworzenia tuneli. W przypadku kompilacji ze źródeł obydwa programy znajdują się w źródłach paczki *ipsec-tools*. *racoon* jest demonem zajmującym się procesem IKE oraz ustanawianiem właściwego SA. *setkey* natomiast służy do zarządzania zawartością SAD i SPD.

Przystępujemy do konfiguracji IPSec na naszych bramkach. W katalogu `/etc/racoon` potrzebne będą cztery pliki:

- `backupsa.txt` - w którym *racoon* przechowuje SA, dopisując kolejne do niego.
- `psk.txt` - zawiera pre-shared keys, czyli klucze wykorzystywane do fazy pierwszej IKE (w tym tekście nie ma mowy o certyfikatach X509)
- `racoon.conf` - konfiguracja demona *racoon*, a w niej między innymi opis SA
- `ipsec.conf` - plik, za pomocą którego będziemy dodawać dodatkowe SP

Plik `ipsec.conf` jest tak naprawdę skrypcem, który interpretowany jest przez program *setkey*. Ma on za zadanie utworzenie odpowiednich wpisów w SPD. Oto zawartość pliku `ipsec.conf` dla naszego przykładu w lokalizacji centralnej B:

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

# Czyszcimy bazy SAD i SPD.
flush;
spdf flush;

# Definicja SA do i z "A".

spdadd 10.2.0.0/24 10.1.0.0/24 any -P out ipsec
      esp/tunnel/172.16.31.5-192.168.0.1/require;

spdadd 10.1.0.0/24 10.2.0.0/24 any -P in ipsec
```

```
    esp/tunnel/192.168.0.1-172.16.31.5/require;

# Definicja SA do i z "C".

spdadd 10.2.0.0/24 10.3.0.0/24 any -P out ipsec
    esp/tunnel/172.16.31.5-192.168.10.34/require;

spdadd 10.3.0.0/24 10.2.0.0/24 any -P in ipsec
    esp/tunnel/192.168.10.34-172.16.31.5/require;
```

W pliku na początku czyszcimy bazę SAD za pomocą polecenia flush, a następnie poleceniem spdflush czyszcimy bazę SPD. Dodajemy kolejne SP, pamiętając o tym, że konieczne jest dodawanie osobnego SP dla każdego kierunku połączenia IPsec. Definiujemy tu kolejno sieci, które będą podlegać danej SP, protokoły które będą jej podlegały (any), kierunek (out lub in) i ipsec jako protokół, któremu poddane zostaną wskazane pakiety. Następnie określamy jak dany pakiet ma być przetworzony. W naszym przypadku będzie to esp w trybie tunelu, którego końce należy określić. require oznacza, że dla tych pakietów wymagane jest utworzone SA.

W ten sposób utworzone zostały połączenia z punktu centralnego do naszych dwu lokalizacji.

Odpowiedni plik ipsec.conf dla lokalizacji A będzie wyglądał następująco:

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

# Czyszcimy bazy SAD i SPD.
flush;
spdflush;

# Definicja SA do i z "B".

spdadd 10.1.0.0/24 10.2.0.0/24 any -P out ipsec
    esp/tunnel/192.168.0.1-172.16.31.5/require;

spdadd 10.2.0.0/24 10.1.0.0/24 any -P in ipsec
    esp/tunnel/172.16.31.5-192.168.0.1/require;
```

Pozostaje skonfigurowanie demona racoon. W tym celu w pliku racoon.conf w lokalizacji B należy wprowadzić następujące ustawienia:

```
# racoon.conf
path pre_shared_key "/etc/racoon/psk.txt" ;

path backupsa "/etc/racoon/backupsa.txt" ;

# Sekcja remote określa parametry dla fazy pierwszej czyli ISAKMP.

# Adres drugiego końca tunelu do lokalizacji A.
remote 192.168.0.1
{
    exchange_mode main;
    proposal {
        encryption_algorithm des;
```

```

        hash_algorithm md5;
        authentication_method pre_shared_key ;
        dh_group modp1024;
    }
}

sainfo address 10.0.2.0/24 any address 10.0.1.0/24 any
{
    pfs_group modp768;
    encryption_algorithm des ;
    authentication_algorithm hmac_md5;
    compression_algorithm deflate;
}

# I odpowiednio dla drugiego tunelu do lokalizacji C.
remote 192.168.10.34
{
    exchange_mode main;
    proposal {
        encryption_algorithm des;
        hash_algorithm md5;
        authentication_method pre_shared_key ;
        dh_group modp1024;
    }
}

# Poniżej definiowane są parametry dla fazy drugiej,
# czyli ustanawiania połączeń SA.
sainfo address 10.2.0.0/24 any address 10.3.0.0/24 any
{
    pfs_group modp768;
    encryption_algorithm des ;
    authentication_algorithm hmac_md5;
    compression_algorithm deflate;
}

```

Oczywiście, odpowiednie pliki racoon.conf powinny się znaleźć w hostach w lokalizacjach A i C. Ich konstrukcja będzie praktycznie identyczna, więc pozostawiam ich stworzenie Czytelnikowi.

I to w zasadzie jest wszystko. Warto jeszcze pamiętać o tym żeby na firewallu odblokować protokoły IPSEC-ESP i IPSEC-AH, które są opisane w /etc/protocols:

```

esp      50      IPSEC-ESP      # Encap Security Payload [RFC2046]
ah       51      IPSEC-AH       # Authentication Header [RFC2402]

```

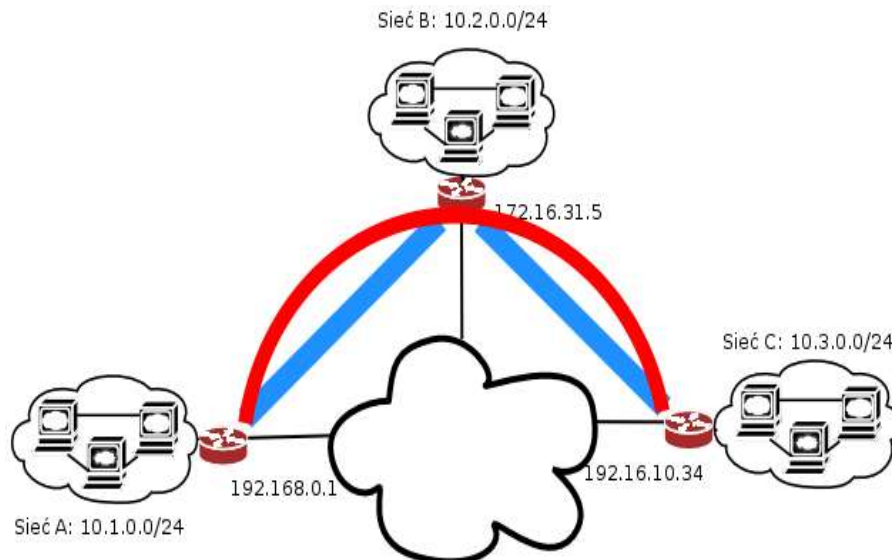
I dodatkowo: port 500 dla protokołu UDP. Całość można zrobić jak poniżej:

```

# Wpuszczamy IPSEC-ESP
iptables -A INPUT -i eth0 -p 50 -j ACCEPT
# Wpuszczamy IPSEC-ESP
iptables -A INPUT -i eth0 -p 51 -j ACCEPT
# Komunikacja ISAKMP
iptables -A INPUT -i eth0 -p 17 --destination-port 500 -j ACCEPT

```

Aby sytuację nieco skomplikować, dodajmy jeszcze jedno zadanie dla naszych tuneli IPsec. Nie dość, że mają one mieć połączenie do centralnej sieci firmowej, ale dodatkowo chcielibyśmy aby poszczególne biura widziały się wzajemnie. Może się to nie do końca wydawać oczywiste, że takie połączenie nie zostało jeszcze stworzone, ale przecież - jak już wcześniej zostało wspomniane - SPD zawiera dokładne wytyczne dotyczące tego co ma być zaakceptowane jako dane, które mają być tunelowane. W SA, które wcześniej zostały opisane w konfiguracji, są określone jedynie pary A-B, B-A i A-C, C-A. Dlatego należy dodatkowo w SA opisać tunel A-C i C-A.



Rys. 3 Dodatkowy tunel pomiędzy lokalizacjami A i C został oznaczony czerwonym kolorem.

Ostatecznie okazuje się to bardzo proste. Potrzeba jedynie zaktualizować pliki ipsec.conf definiując dodatkowe SP w bazie SPD. W centralnym hoście należy wskazać co robić z pakietami z sieci C do A i odwrotnie.

```
# Dodatkowe ustawienia do polaczenia lokalizacji A i C.
spdadd 10.1.0.0/24 10.3.0.0/24 any -P in ipsec
        esp/tunnel/192.168.10.34-172.16.31.5/require;
spdadd 10.1.0.0/24 10.3.0.0/24 any -P out ipsec
        esp/tunnel/172.16.31.5-192.168.0.1/require;

spdadd 10.3.0.0/24 10.1.0.0/24 any -P in ipsec
        esp/tunnel/192.168.10.34-172.16.31.5/require;
spdadd 10.3.0.0/24 10.1.0.0/24 any -P out ipsec
        esp/tunnel/172.16.31.5-192.168.0.1/require;
```

Podobnie w plikach ipsec.conf w oddziałach trzeba wskazać żeby pakiety skierowane do odległych oddziałów były przekazywane i szyfrowane przez IPsec. Poniżej przykład dla lokalizacji C.

```
# Definicja polaczenia z C do A.
spdadd 10.1.0.0/24 10.3.0.0/24 any -P in ipsec
        esp/tunnel/172.16.31.5-192.168.10.34/require;
```

```
spdadd 10.3.0.0/24 10.1.0.0/24 any -P out ipsec
      esp/tunnel/192.168.10.34-172.16.31.5/require;
```

I dodatkowo odpowiedni wpis na bramce IPsec z drugiej strony, który także pozostawiam Czytelnikowi.

Po uruchomieniu demona racoon.conf oraz pliku ipsec.conf, (który tak naprawdę jest skryptem wykonywanym przez setkey) i próbie połączenia się przez tunel, w logach powinniśmy zobaczyć szczegółowe informacje o procesie zestawiania tunelu IPsec, jak widać poniżej. Warto tylko zwrócić uwagę na zestawione dwa SA, które bardzo dobrze widać w czterech ostatnich liniach. Jest to zapis zestawiania połączenia IPsec od samego startu demona racoon z punktu lokalizacji centralnej.

```
INFO: main.c:174:main(): @(#)racoon - IPsec-tools 0.2.3
INFO: main.c:175:main(): @(#)This product linked OpenSSL 0.9.7c 30 Sep 2003
(http://www.openssl.org/)
INFO: isakmp.c:1375:isakmp_open(): 172.16.31.5[500] used as isakmp port (fd=9)
INFO: isakmp.c:1701:isakmp_post_acquire(): IPsec-SA request for 192.168.0.1 queued due to no
phases found.
INFO: isakmp.c:795:isakmp_ph1begin_i(): initiate new phase 1 negotiation: 172.16.31.5[500]
<=>192.168.0.1[500]
INFO: isakmp.c:800:isakmp_ph1begin_i(): begin Identity Protection mode.
INFO: isakmp.c:2431:log_ph1established(): ISAKMP-SA established 172.16.31.5[500]-192.168.0.1
[500] spi:4e32d9efaba23dcc:fb440e9e187358f6
INFO: isakmp.c:1046:isakmp_ph2begin_r(): respond new phase 2 negotiation: 172.16.31.5[0]
<=>192.168.0.1[0]
INFO: isakmp.c:939:isakmp_ph2begin_i(): initiate new phase 2 negotiation: 172.16.31.5[0]
<=>192.168.0.1[0]
INFO: pfkey.c:1127:pk_recvupdate(): IPsec-SA established: ESP/Tunnel 192.168.0.1-
>172.16.31.5 spi=178411656(0xaa25888)
INFO: pfkey.c:1348:pk_recvadd(): IPsec-SA established: ESP/Tunnel 172.16.31.5->192.168.0.1
spi=109621173(0x688afb5)
INFO: pfkey.c:1127:pk_recvupdate(): IPsec-SA established: ESP/Tunnel 192.168.0.1-
>172.16.31.5 spi=178411656(0xaa25888)
INFO: pfkey.c:1348:pk_recvadd(): IPsec-SA established: ESP/Tunnel 172.16.31.5->192.168.0.1
spi=111486393(0x6a525b9)
```

UFF... KONIEC:)

Podane tu przykłady stanowią jedynie muśnięcie tematu jakim jest IPsec. Sama lektura manuali dla setkey, racoon, racoon.conf i ipsec.conf daje pojęcie o tym jak ciekawy i zarazem złożony jest IPsec. Dla ułatwienia poznawania go bardzo polecam artykuł Pawła Krawczyka³ i IPsec-howto⁴ (dzięki którym powstał także ten tekst). Po tych lekturach RFC staje się znośniejsze do czytania, a sama konfiguracja szyfrowanych tuneli jest znacznie bardziej klarowna.

Można na koniec zadać sobie jeszcze pytanie, czy gra jest warta świeczki, czy skomplikowanie IPsec nie jest zbyt duże, kiedy są pod ręką znacznie prostsze w konfiguracji możliwości tworzenia tuneli w oparciu o PPTP, L2TP czy choćby OpenVPN. Być może, ale faktem jest, że IPsec stał się standardem tworzenia połączeń VPN, zaakceptowanym przez największych producentów urządzeń sieciowych takich jak Cisco, Juniper, czy tych z niższej półki, jak np. Zyxel i Linksys. Jeśli urządzenie tych producentów będzie miało możliwość tworzenia VPN, najprawdopodobniej będzie to w oparciu o IPsec, co oznacza możliwość łączenia urządzeń różnych producentów i naszych bramek IPsec pracujących pod Linuxem. A to znacznie upraszcza łączenie

3 Świetne źródło informacji o IPsec po polsku - <http://www.ipsec.pl/ipsec/>

4 The official IPsec Howto for Linux - <http://www.ipsec-howto.org/>

różnych sieci w jedną całość, oczywiście – bezpieczną całość;)

LINKI:

<http://www.ipsec.pl/ipsec>
<http://www.ipsec-howto.org/>
<http://www.kame.net/>
RFC 2411, RFC 2401