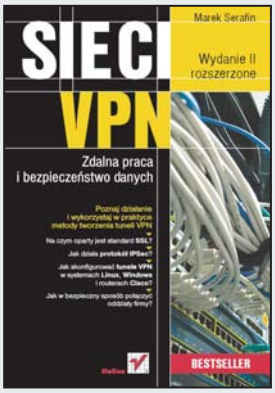
Sieci VPN. Zdalna praca i bezpieczeństwo danych. Wydanie II rozszerzone



Autor: Marek Serafin
ISBN: 978-83-246-2474-4
Format: 158×235, stron: 248

Poznaj działanie i wykorzystaj w praktyce metody tworzenia tuneli VPN

- Na czym oparty jest standard SSL
- Jak działa protokół IPSec
- Jak skonfigurować tunele VPN w systemach Linux, Windows i routerach Cisco?
- Jak w bezpieczny sposób połączyć oddziały firmy?

Rozwój sieci pozwolił wielu firmom i organizacjom na sprawną i szybką komunikację i tym samym otworzył nowe perspektywy dla tych pracowników, którzy z różnych względów wykonują swoje obowiązki poza biurem. Niestety – zdalny dostęp do firmowej infrastruktury IT niesie ze sobą także zagrożenia związane z możliwością utraty, uszkodzenia lub wydostania się na zewnątrz cennych danych. Rozwiązaniem tego problemu są łącza szyfrowane, nazywane VPN. Jeśli chcesz stworzyć właśnie taką możliwość bezpiecznej choć zdalnej pracy w Twojej firmie, z tego podręcznika dowiesz się jak to zrobić!

Książka „Sieci VPN. Zdalna praca i bezpieczeństwo danych. Wydanie II" to drugie, poszerzone wydanie praktycznego przewodnika dla administratorów sieci firmowych, którzy zajmują się wdrażaniem rozwiązań umożliwiających pracę na odległość. Ten bestsellerowy podręcznik opisuje wszystkie aspekty konfigurowania tuneli VPN z wykorzystaniem protokołów IPSec oraz SSL w systemach Linux, Windows oraz routerach Cisco. Czytając ją poznasz standard SSL, zasady generowania certyfikatów, a także zrozumiesz zasadę działania protokołu IPSec. Analizując zawarte w książce przykłady, nauczysz się otwierać zdalny dostęp do sieci korporacyjnej, łączyć oddziały firmy za pomocą IPSec i uruchamiać tunele VPN w urządzeniach mobilnych.

- Zagrożenia wynikające z konstrukcji protokołu TCP/IP
- Przesyłanie danych z wykorzystaniem SSL
- Generowanie kluczy i certyfikatów
- Zapewnianie pracownikom zdalnego dostępu do zasobów firmy
- Instalacja i konfiguracja programu OpenVPN
- Tunele VPN w urządzeniach mobilnych
- Implementacja IPSEC/L2TP w systemie Linux
- Konfiguracja IPSec w routerach i Cisco
- Konfiguracja VPN w systemach Windows Server
- Metody łączenia oddziałów firmy w systemach Linux, Cisco, Windows

Otwórz bezpieczny, zdalny dostęp do sieci!

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Rozdział 7.

IPSec w systemie Linux

7.1. IPSec — przygotowanie środowiska w systemie Linux

W niniejszym punkcie omówię przygotowanie systemu do działania z protokołem IPSec. Najprawdopodobniej nie obejdzie się bez ręcznej kompilacji niektórych składników, dlatego też zakładam, że używasz jakiejś aktualnej wersji Linuksa oraz masz zainstalowane narzędzia do kompilacji programów (`gcc`, `make` itd.).

Z punktu widzenia systemu operacyjnego połączenie IPSec można podzielić na dwie części:

1. Część odpowiedzialną za zarządzanie pakietami (protokół AH/ESP) — tj. enkapsulację pakietów IP w pakiety IPSec, zabezpieczanie sum kontrolnych itd. Z racji tego, że operacje te muszą być bardzo wydajne, ich obsługą zajmuje się jądro systemu.
2. Część odpowiedzialną za zestawienie połączenia i późniejszą wymianę kluczy (protokół IKE). Obsługą tych funkcji zajmuje się program (demon) działający w warstwie użytkownika — w systemie Linux to demon Pluto (wchodzący w skład OpenSWAN).

W związku z powyższym jądro systemu musi obsługiwać protokół IPSec. Dla systemu Linux powstały dwa niezależne stosy obsługi IPSec — starszy KLIPS oraz nowszy NETKEY. Stos KLIPS, rozwijany od prawie dziesięciu lat, dobrze sprawdzony, działa z jądrami serii 2.2, 2.4, 2.6. Niestety, kod stosu KLIPS nie wchodzi w skład źródeł jądra Linuksa, dlatego konieczne jest nałożenie łatek na źródła i przekompilowanie jądra. Ze stosem KLIPS jest także problem w przypadku połączeń zza NAT-a — aby udało się zestawić takie połączenie IPSec, potrzebna jest jeszcze jedna łatka — NAT-Traversal. Wspomnianych wad nie posiada stos NETKEY, dostępny początkowo tylko dla jąder 2.6, a obecnie także dla 2.4. W przykładach używać będziemy stosu NETKEY.

W popularnych dystrybucjach systemu Linux domyślnie dostarczane są wszystkie potrzebne moduły. Sprawdź poleceniem `find`, czy rzeczywiście masz w systemie odpowiednie pliki. Wpisz polecenie:

```
root@server:~# find /lib/modules/`uname -r` -name 'esp4*'
```

Program *find* powinien odnaleźć plik modułu jądra o nazwie `esp4.ko` lub `esp4.o`. Jeśli plik istnieje, oznacza to, że twórcy dystrybucji dodali moduły IPSec. Jeśli kompilowałeś kernela z własnymi opcjami i nie masz modułów IPSec, będziesz musiał przekompilować jądro od nowa. Poniżej podaję listę opcji, które musisz dodać jako moduły. Nazwy w nawiasach kwadratowych to dokładne nazwy z pliku konfiguracyjnego jądra `.config`.

Uruchom program `make menuconfig` i wybierz poniższe opcje konfiguracyjne:

W gałęzi Networking --> Networking options ---> wybierz
```
IPsec user configuration interface (NEW) [CONFIG_XFRM_USER]
PF_KEY sockets [CONFIG_NET_KEY]
IP: AH transformation   [CONFIG_INET_AH]
IP: ESP transformation [CONFIG_INET_ESP]
IP: IPComp transformation [CONFIG_INET_IPCOMP]
IP: IPsec transport mode [CONFIG_INET_XFRM_MODE_TRANSPORT]
IP: IPsec tunnel mode  [CONFIG_INET_XFRM_MODE_TUNNEL]
IP: advanced router [CONFIG_IP_ADVANCED_ROUTER]
Packet socjet [CONFIG_PACKETT]
```

W gałęzi Device Drivers ---> Character devices ---> wybierz:
```
Legacy (BSD) PTY support [CONFIG_LEGACY_PTYS]
```

W gałęzi Device Drivers ---> Network device support ---> wybierz:
```
PPP (point-to-point protocol) support [CONFIG_PPP]
PPP support for sync tty ports [CONFIG_PPP_SYNC_TTY]
PPP Deflate compression [CONFIG_PPP_DEFLATE]
PPP BSD-Compress compression [CONFIG_PPP_BSDCOMP]
```

Zapisz zmiany w pliku konfiguracyjnym jądra, a następnie przekompiluj jądro.

7.2. Instalacja programu OpenSWAN

OpenSWAN to implementacja protokołu IPSec w systemie Linux rozwijana przez grupę deweloperów, którzy wcześniej pracowali nad projektem FreeSwan. W trakcie prac nad projektem FreeSwan doszło do konfliktu i część deweloperów rozpoczęła pracę nad OpenSWAN. FreeSwan nie jest już zresztą rozwijany i nie zalecam używania go (swego czasu był bardzo niestabilny, czego osobiście doświadczyłem).

Jeżeli Twoja dystrybucja wspiera automatyczną instalację pakietów, możesz zainstalować program OpenSWAN z gotowych paczek. W przypadku Debiana możesz zainstalować pakiet, używając programu `apt`.

Instalacja programu ze źródeł przebiega następująco:

1. Ściągnij źródła programu OpenSWAN ze strony projektu: *http://www.openswan.org/*.
2. Sprawdź podpis cyfrowy pakietu.
3. Rozpakuj archiwum i wpisz polecenie

```
make programs install
```

Jeżeli podczas kompilacji programu zostanie zgłoszony błąd, upewnij się, czy masz zainstalowany pakiet z nagłówkami biblioteki arytmetycznej GMP (GNU Multiple Precision Arithmetic Library). W przypadku dystrybucji Debian odpowiedni pakiet nazywa się `libgmp3-dev`. Dla innych dystrybucji paczka powinna nazywać się podobnie — ważne, aby był to pakiet „deweloperski", tzn. zawierał pliki nagłówkowe.

Jeśli kompilacja przebiegnie prawidłowo, w systemie zostanie zainstalowany program zarządzający połączeniami IPSec (*/usr/local/sbin/ipsec*) oraz demon protokołu IKE — Pluto.

Ostatnim (opcjonalnym) składnikiem, który musisz zainstalować, jest serwer L2TP — najlepiej xl2tpd. Napisałem: opcjonalnym, gdyż nie wszystkie rozwiązania IPSec używają tunelowania L2TP. Niemniej implementacja IPSec firmy Microsoft wbudowana w każdy z systemów Windows wymaga do działania właśnie protokołu L2TP. Jeżeli zamierzasz łączyć się z bramą VPN, używając wbudowanych w Windows mechanizmów IPSec, musisz zainstalować i skonfigurować demon L2TP.

Protokół L2TP umożliwia przesyłanie ramek połączenia PPP poprzez protokół IP (internet), które to połączenie normalnie realizowane jest tylko w bezpośrednim połączeniu punkt-punkt (modemy, linie szeregowe itd.). Samo połączenie PPP operuje w warstwie drugiego modelu OSI i służy do enkapsulacji protokołów warstwy wyższej (IP, IPX itd.), zapewniając jednocześnie uwierzytelnianie oraz kompresję. Połączenie protokołu L2TP z PPP umożliwia tunelowanie protokołu IP w ramach innego połączenia IP, dlatego często wykorzystywane jest w sieciach VPN. Wykorzystanie protokołu PPP daje także dodatkowe możliwości, jak np. przydzielanie adresów IP tunelowi, przekazywanie parametrów sieciowych, takich jak DNS, WINS itp.

Aby zainstalować demon L2TP, wykonaj następujące czynności:

1. Pobierz źródła pakietu ze strony *http://www.xelerance.com/software/xl2tpd/*.
2. Sprawdź podpis pakietu.
3. Rozpakuj archiwum i wpisz komendę

```
make
```

   Z kompilacją tego programu nie powinno być problemów.

4. Wpisz polecenie `make install`, aby zainstalować skompilowany program we właściwych katalogach (*/usr/sbin/xl2tpd*).

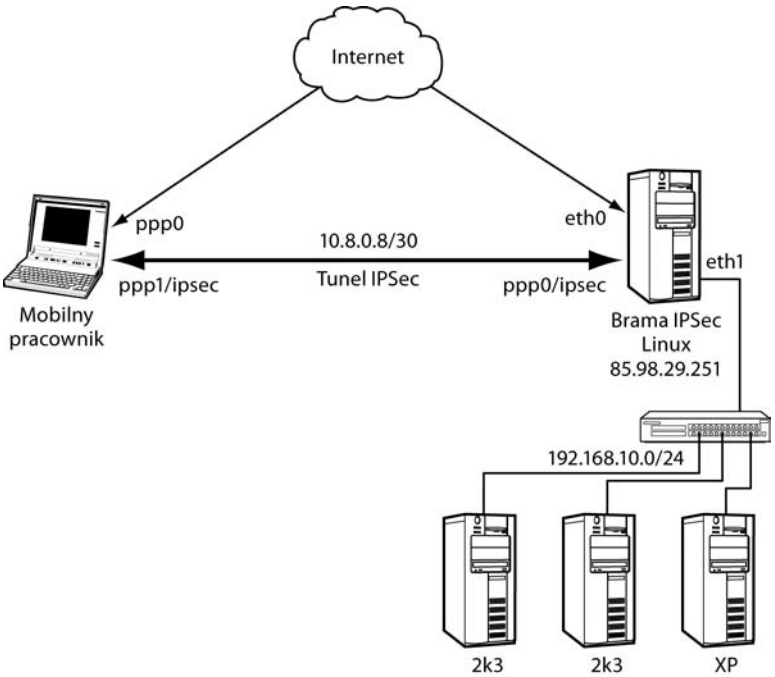
Naturalnie aby tunelowanie połączenia PPP przez protokół L2TP mogło działać, potrzebny jest także demon PPP (sprawdź obecność polecenia `pppd`). Jednakże pakiet ten instalowany jest chyba we wszystkich dystrybucjach Linuksa.

7.3. Praktyczny przykład — brama IPSec/VPN dla użytkowników mobilnych

W niniejszym punkcie stworzymy bramę VPN dla użytkowników mobilnych łączących się zdalnie z siedzibą firmy. Użytkownicy używają na laptopach systemu Microsoft Windows XP i wbudowanego weń „klienta" IPSec. Analogicznie jak w przykładzie z rozdziału 5. (brama OpenVPN), użytkownicy powinni mieć dostęp do kilku serwerów w sieci LAN. Adresy IP użytkowników mobilnych nie są znane, mogą oni łączyć się z różnych miejsc, także zza maskarady (połączenia GPRS, hotspot itd.). Po stronie bramy VPN wykorzystamy system Linux (jądro 2.6 ze stosem IPSec NETKEY) oraz oprogramowanie OpenSWAN.

Konfigurację przedstawia rysunek 7.3.1.

Rysunek 7.3.1. *Brama IPSec dla użytkowników mobilnych*




Uwaga

W tym przypadku do uwierzytelniania użytkowników powinniśmy użyć certyfikatów, jednakże w pierwszej kolejności wykorzystamy klucz współdzielony, a następnie rozbudujemy konfigurację o certyfikaty. Zalecam stosowanie takiej kolejności, gdyż konfiguracja oparta na kluczu współdzielonym jest prostsza, a ponadto warto przetestować połączenie IPSec, eliminując wszelkie możliwe komplikacje. Jest to szczególnie ważne z tego powodu, że instalacja certyfikatów po stronie systemów Windows jest trochę skomplikowana (szczegóły w następnych punkach rozdziału).

7.3.1. Konfiguracja bramy IPSec (Linux)

Zainstaluj oprogramowanie OpenSWAN oraz demona xl2tpd zgodnie z opisem zawartym w poprzednim punkcie.

Konfigurację zaczniemy od przygotowania demona L2TP. Utwórz plik konfiguracyjny */etc/l2tp/l2tpd.conf*.

Zawartość pliku konfiguracyjnego *l2tpd.conf* powinna wyglądać następująco (patrz listing 7.3.1.1):

Listing 7.3.1.1. *Plik konfiguracyjny demona L2TP*

```
[global]
 listen-addr = 85.98.29.251                    ;adres internetowy bramy
 port = 1701                                   ;port — zostaw domyślny

[lns default]
ip range = 192.168.10.198-192.168.10.250       ;pula IP dla klientów
local ip = 85.98.29.251                        ;IP lokalny połączenia PPP
require chap = yes                             ;wymagamy uwierzytelniania CHAP
;refuse pap = yes
require authentication = yes
name = ipsec
ppp debug = yes
pppoptfile = /etc/ppp/options.l2tpd            ;pozostały
length bit = yes
```

Zauważ, że wymóg uwierzytelniania dotyczy tutaj połączenia PPP, a nie L2TP. Demon L2TP ma możliwość przeprowadzania swojego uwierzytelniania, ale nie jest to w tym przypadku potrzebne.

Teraz utwórzmy plik z opcjami programu `pppd` — */etc/ppp/options.l2tpd*. Przykładową konfigurację przedstawiono na listingu 7.3.1.2.

Listing 7.3.1.2. *Konfiguracja połączenia PPP*

```
ipcp-accept-local          ; pppd zaakceptuje lokalny adres połączenia
ipcp-accept-remote         ; jw. dla adresu drugiej strony
require-mschap-v2          ; wymagamy uwierzytelniania MSCHAP wersji drugiej
auth
proxyarp
idle 1800
mtu 500
mru 500
# eof
```

Następnie konfigurujemy plik z danymi uwierzytelniającymi dla PPP — */etc/ppp/chap-secrets*

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses
marek      *    "test"            *
```

gdzie `marek` to nazwa użytkownika, a `"test"` — hasło.

W sytuacji gdyby PPP było jedynym uwierzytelnianiem, mógłbyś wpisać w pliku `chap-secrets` wszystkich użytkowników. Nie ma to jednak sensu, gdyż podstawowym uwierzytelnianiem będą certyfikaty X.509. Niemniej konfigurację PPP można wykorzystać do przypisywania stałych adresów IP dla poszczególnych klientów, np.:

```
jacek  *       "test12"  192.168.10.220
michal  *       "test12"  192.168.10.221
```

Ostatni element przygotowania bramy IPSec to właściwa konfiguracja programu OpenSWAN, która sprowadza się w najprostszym przypadku do edycji dwóch plików — *ipsec.conf* oraz *ipsec.secrets*.

Na listingu 7.3.1.3 przedstawiony został plik *ipsec.conf*. W tym przypadku metodą uwierzytelniania jest klucz współdzielony, a klientami mogą być użytkownicy Windowsa (XP, Vista) znajdujący się za NAT-em.

Listing 7.3.1.3. *Konfiguracja programu OpenSWAN z użyciem PSK*

```
version 2.0
config setup
        interfaces=%defaultroute
        plutodebug=none
        forwardcontrol=yes
        nat_traversal=yes

virtual_private=%v4:10.0.0.0/8,%v4:172.16.0.0/12,%v4:192.168.0.0/16,%v4:!192.168.10.0/24

conn roadwarrior-l2tp
        leftprotoport=17/1701
        rightprotoport=17/1701
        also=roadwarrior

conn roadwarrior
        auth=esp
        authby=secret
        compress=yes
        keyexchange=ike
        keyingtries=3
        pfs=no
        rekey=yes
        left=%defaultroute
        right=%any
        rightsubnet=vhost:%no,%priv
        auto=add

#Disable Opportunistic Encryption
include /etc/ipsec.d/examples/no_oe.conf
```

Twórcy programu OpenSWAN przyjęli konwencję, aby stron połączenia nie nazywać w klasyczny sposób: „serwer” i „klient” (lub też „źródło” i „cel”), tylko „strona lewa” (ang. *left*) i „strona prawa” (ang. *right*). Ma to swoje uzasadnienie, ponieważ tak naprawdę i tak zestawiane są dwa niezależne skojarzenia SA — dla każdego z kierunków transmisji. Poza tym nie zawsze jest jasne, która strona jest bardziej serwerem, a która klientem (przykład łączenia oddziałów firm). Przez „left” rozumie się lokalną konfigurację,

a przez „right” odległą stronę. Zauważ, że w przypadku połączeń typu użytkownik mobilny-brama IPSec, patrząc od strony bramy IPSec, strona „right” najczęściej nie jest znana (nie znamy adresu IP, z którego będzie się łączyć użytkownik). W przypadku łączenia dwóch bram IPSec (łączenie oddziałów firmy) strony „left” i „right” są na ogół ściśle określone. W anglojęzycznej terminologii pracowników mobilnych, którzy łączą się z różnych miejsc kraju i świata, przyjęło się nazywać *roadwarrior* (polskim odpowiednikiem będą „jeżdżący handlowcy w autach z kratką” :-)). Wspominam o terminie *roadwarrior*, ponieważ bardzo często występuje on w dokumentacji technicznej różnych implementacji VPN.

Przeanalizujmy najważniejsze wpisy pliku konfiguracyjnego z listingu 7.3.1.3.

`version 2.0` — informuje program OpenSWAN, że składnia pliku będzie zgodna z OpenSWAN, a nie z przestarzałym programem FreeSwan (poprzednik OpenSWAN).

Plik podzielony jest na kilka sekcji. Podstawowe opcje — tzw. globalne — zawarte są w sekcji `config setup`.

`interfaces=%defaultroute` — oznacza interfejs, na którym ma działać IPSec. Wartość domyślna to `%defaultroute`, co oznacza, że użyty zostanie „wyjściowy” adres IP komputera (z którego komputer „wychodzi na świat”).

Na ogół `%defaultroute` jest poprawną wartością.

`plutodebug=none` — definiuje poziom szczegółowości (ang. *verbose level*) demona `pluto`. Opcja przydaje się podczas debugowania, gdy nie działa połączenie. W przypadku braku jawnie podanej opcji domyślnie przyjmowana jest wartość `none`. Inne możliwe wartości to: `all`, `raw`, `krypt`, `parsing`, `emitting`, `control`.

`all` — bardzo szczegółowe debugowanie. Opcja przydatna dla guru IPSec. Zwykli śmiertelnicy mogą mieć problemy ze zrozumieniem pojawiających się komunikatów (jakkolwiek czasem może się przydać).

Podczas debugowania polecam opcję `control`, a gdy wszystko zacznie działać — `none`.

`forwardcontrol=yes|no` — sprawdza, czy załączone jest przekazywanie pakietów IP (ang. *IP forwarding*). Jeżeli nie, to je załącza. Po zakończeniu działania tunelu przywraca poprzednią wartość.

`nat_traversal=yes|no` — domyślnie: `no`. Ważna opcja, jeżeli spodziewasz się połączeń zza NAT-a (maskarady). Załączenie jej sprawi, że OpenSWAN będzie oczekiwał także połączeń na porcie UDP 4500, po którym przenoszone są pakiety IPSec.

`virtual_private=` — określa podsieci (z zakresu „prywatnych” klas IP), z których mogą łączyć się klienci. Na ogół podaje się tutaj wszystkie pule IP zdefiniowane do użytku prywatnego, z wyjątkiem puli używanej w sieci firmowej, do której użytkownicy chcą mieć dostęp poprzez VPN. Innymi słowy — OpenSWAN potrzebuje znać adres IP klienta. Jako że użytkownicy mogą łączyć się zewsząd, a my nie znamy prywatnych klas adresowych używanych w hotelach, hotspotach itd., podajemy wszystkie „prywatne” pule adresowe. Z oczywistych względów (problem z routingiem) musimy wykluczyć

pulę używaną u nas w firmie. Jeżeli mielibyśmy pewność, że wszyscy użytkownicy będą łączyć się zza NAT-u, z puli adresowej `192.168.1.0/24` — to wystarczyłoby podać tylko tę pulę. Zwróć też uwagę na parametr `rightsubnet` opisany w dalszej części.

Sekcja `conn roadwarrior-l2tp` odpowiedzialna jest za połączenia L2TP (port UDP 1701). Jest ona potrzebna, jeśli oczekujesz połączeń od klientów wbudowanych w systemy Windows (używają L2TP). Ważne jest, aby sekcja L2TP umieszczona była w pliku przed właściwą sekcją odpowiedzialną za połączenia klientów (u nas: `conn roadwarrior`).

`conn roadwarrior` — właściwa sekcja połączeń dla pracowników mobilnych.

`auth=esp` — określa protokół IPSec. Możliwe opcje to `ah` lub `esp`. Zalecaną metodą jest ESP, ponieważ obsługuje uwierzytelnianie i szyfrowanie.

`authby=secret` — określa sposób uwierzytelniania stron. Wartość `secret` oznacza współdzielony klucz (PSK). W przypadku uwierzytelniania z wykorzystaniem certyfikatów opcja powinna mieć wartość `rsasig`.

`compress=yes` — możliwa kompresja danych.

`keyexchange=ike` — wartość `ike` oznacza, że do uzgodnienia kluczy zostanie użyty protokół IKE (Internet Key Exchange). Użycie IKE jest zdecydowanie polecane (także przez twórców OpenSWAN). Ręczne zarządzanie tunelami jest skomplikowane i niepraktyczne.

`keyingtries=3` — określa, ile prób negocjacji SA może nastąpić (maksymalnie).

`pfs=yes|no` — włącza (`yes`) lub wyłącza (`no`) PFS.

`rekey=yes` — określa, czy połączenie po wygaśnięciu powinno być renegocjowane. Możliwe wartości to `yes` lub `no`. Domyślnie: `yes`.

`right=%any` — określa adres drugiej strony. Słowo kluczowe `%any` oznacza, że adres IP nie jest znany (przypadek mobilnych pracowników).

`rightsubnet=vhost:%no,%priv` — opcja `rightsubnet` określa podsieć drugiej strony. W przypadku połączeń typu `Roadwarriors` z możliwym NAT-em oraz „nie NAT-em" powinna mieć wartość `vhost:%no,%priv`. Przez NAT oraz „nie NAT" rozumiem, że użytkownicy mogą łączyć się zza NAT-u lub też mieć „zewnętrzny" adres IP i obie konfiguracje będą działać jednocześnie.

Więcej szczegółów dotyczących tego zagadnienia znajdziesz na stronie *http://www.openswan.org/docs/local/README.NAT-Traversal*.

`auto=add` — opcja przyjmuje wartości: `start`, `add`, `ignore` (domyślna!) oraz `manual`.

Znaczenie poszczególnych wartości jest następujące:

- `start` — załaduj konfigurację i inicjuj połączenie z drugą stroną. Wartość najczęściej używana w przypadku połączeń dwóch routerów lub jeśli strona jest klientem (ma inicjować połączenie z drugą stroną).

- `add` — załaduj konfigurację i odpowiadaj na przychodzące połączenia (czekaj na połączenie od drugiej strony). Wartość używana dla konfiguracji typu roadwarriors — nie znamy ani czasu, ani adresu IP, z jakiego połączy się mobilny pracownik. Jedyne, co możemy zrobić, to odpowiedzieć na jego połączenia.
- `ignore` — ignoruje sekcję tego połączenia. Uwaga: jest to wartość domyślna, dlatego musisz przypisać jakąś wartość sekcjom, które mają działać.
- `manual` — opcja używana przy ręcznej konfiguracji wymiany kluczy (zamiast użycia IKE). Opcja niepolecana. Nie będziemy rozpatrywali tego przypadku. Zainteresowanych odsyłam do dokumentacji programu OpenSWAN.

W ostatniej linijce pliku konfiguracyjnego widzimy dołączony (ang. *include*) plik:

```
#Disable Opportunistic Encryption
include /etc/ipsec.d/examples/no_oe.conf
```

Dołączenie powyższego pliku wyłącza tzw. szyfrowanie oportunistyczne, które w tym przypadku nie jest potrzebne (strony uwierzytelniają się w inny sposób), a pozostawienie go załączonego skutkowałoby serią komunikatów w logach systemowych, mówiących o niemożności sprawdzenia informacji w DNS-ie. Idea szyfrowania oportunistycznego polega na możliwości uwierzytelnienia dowolnych hostów w internecie bez wcześniejszej wymiany tajnego klucza (lub kluczy publicznych). Uwierzytelnienie hosta odbywa się tutaj na podstawie informacji pobranych z bezpiecznych serwerów DNS (ang. *Secure DNS*). Na razie bezpieczne DNS-y nie występują w powszechnym użyciu.

Uwaga

> Ogólna uwaga odnośnie do składni pliku *ipsec.conf*. Pamiętaj, aby pomiędzy sekcjami połączeń była jedna linijka przerwy. Nazwa sekcji nie powinna zaczynać się od spacji czy tabulacji. Opcje w ramach sekcji mogą zaczynać się od tabulacji. Wcześniejsze wersje programu były na to dość wrażliwe, a zdiagnozowanie błędu kosztowało mnie dużo czasu.

Pozostał nam jeszcze do konfiguracji plik */etc/ipsec.secrets*. W przypadku współdzielonego klucza oraz połączeń z nieznanych adresów IP składnia pliku jest następująca:

```
85.98.29.251 %any: PSK "tajnehaslo"
```

gdzie `85.98.29.251` to „zewnętrzny” adres IP lokalnej strony (odpowiednik `%defaultroute`), a `%any` — adres drugiej strony. Zadbaj o to, aby możliwość odczytu pliku miał tylko użytkownik *root* (`chmod 600`).

7.3.2. Uruchomienie tunelu

Aby uruchomić tunel (proces nasłuchiwania), wpisz polecenie:

```
ipsec setup start
```

lub użyj skryptu startowego z dystrybucji:

```
/etc/init.d/ipsec start
```

7.4. Konfiguracja klienta Windows

W tym punkcie zajmiemy się konfiguracją połączenia IPSec w systemach Windows, tak aby ich użytkownicy mogli łączyć się z naszą bramą linuksową skonfigurowaną w punkcie 7.3.1. Metodą uwierzytelniania będzie klucz współdzielony oraz dodatkowe uwierzytelnianie w połączeniu PPP (MS-CHAP v2). Użyjemy wbudowanego w Windowsa klienta IPSec (2000, XP, 2003, Vista).

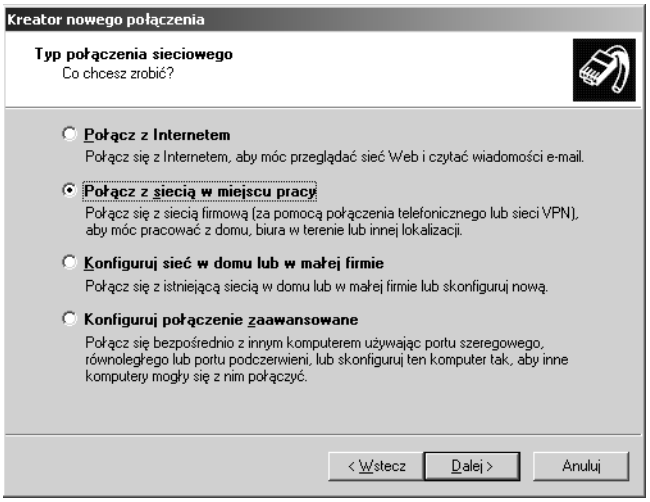

Uwaga

Połączenia z użyciem współdzielonego klucza (PSK) przetestowałem, używając Windowsa XP (SP2). Przeglądając archiwum list dyskusyjnych, wyczytałem, iż konfiguracja w systemach Windows 2000 nie działa prawidłowo z kluczem współdzielonym. Niestety, jako że nie mam już nigdzie „w użyciu" systemu Windows 2000, nie byłem w stanie tego sprawdzić. Konfiguracja z wykorzystaniem certyfikatów działa natomiast prawidłowo. W związku z powyższym jeśli zamierzasz się łączyć, używając systemu Windows 2000, przejdź, proszę, do następnego punktu rozdziału opisującego konfigurację opartą na certyfikatach.

W przypadku Windowsa XP i nowszych sprawa sprowadza się do dodania nowego połączenia przy użyciu kreatora. Aby dodać nowe połączenie, wykonaj poniższe czynności:

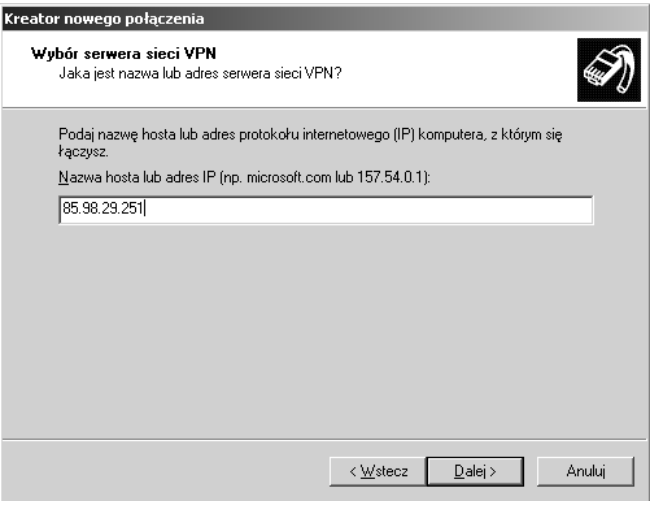
1. Wejdź do *Panelu sterowania* i wybierz *Połączenia sieciowe*.
2. Uruchom *Kreatora nowego połączenia*. Kreator zapyta o rodzaj połączenia — wybierz opcję *Połącz z siecią w miejscu pracy* — tak jak pokazano na rysunku 7.4.1. Następnie kliknij przycisk *Dalej*.

Rysunek 7.4.1. *Kreator nowego połączenia w systemie Windows XP*



3. W następnym kroku wybierz opcję *Połączenie wirtualnej sieci prywatnej*, następnie kliknij przycisk *Dalej*.
4. Wybierz nazwę dla połączenia, np. `ipsec1`.
5. Podaj adres IP lub nazwę DNS bramy VPN (patrz rysunek 7.4.2).

Rysunek 7.4.2.
Kreator połączenia VPN — podaj adres IP bramy IPSec

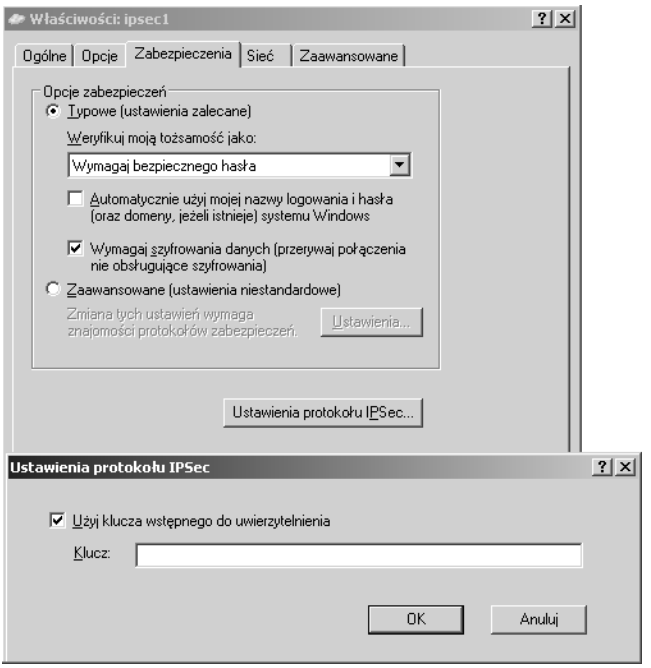


6. Zakończ pracę kreatora, klikając przycisk *Zakończ*. Nowe połączenie powinno widnieć na liście połączeń sieciowych — w grupie *Wirtualna sieć prywatna*.

   Przed uruchomieniem naszego połączenia trzeba zmodyfikować jeszcze dwie opcje. Wejdź we właściwości nowego połączenia — kliknij prawym przyciskiem myszy ikonę nowego połączenia i wybierz z menu opcję *Właściwości*. Postępuj według poniższych punktów.

7. Przejdź do zakładki *Zabezpieczenia* i wybierz opcję *Ustawienia protokołu IPSec* (patrz rysunek 7.4.3).

Rysunek 7.4.3.
Właściwości połączenia IPSec — wpisz hasło (PSK)



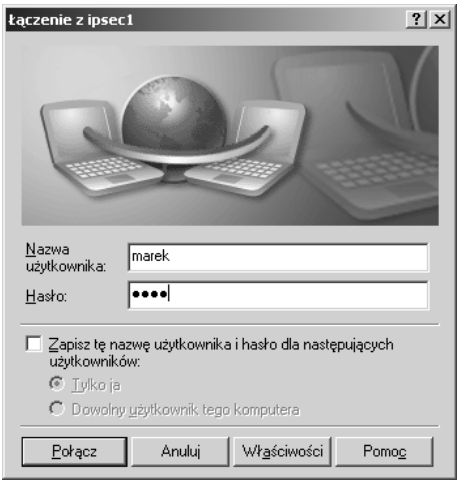
8. Zaznacz opcję *Użyj klucza wstępnego do uwierzytelnienia* oraz wpisz w polu *Klucz* hasło podane w pliku */etc/ipsec.secrets* na Linuksie — w naszym przypadku będzie to: `tajnehaslo`. Zatwierdź przyciskiem *OK*.

9. Następnie przejdź do zakładki *Sieć* i zmień wartość pola *Typ wirtualnej sieci prywatnej (VPN)* z *Automatyczny* na *Sieć VPN z protokołem L2TP IPSec* — patrz rysunek 7.4.4.

Rysunek 7.4.4. *Właściwości połączenia IPSec — wybierz protokół L2TP*

10. Zapisz zmiany.

W tej chwili możesz połączyć się z bramą IPSec. Kliknij dwukrotnie myszką ikonę połączenia VPN (patrz rysunek 7.4.5). W oknie dialogowym podaj nazwę użytkownika i hasło, a następnie kliknij przycisk *Połącz*.



Rysunek 7.4.5. *Pierwsze połączenie IPSec*

Nazwa użytkownika i hasło to oczywiście dane uwierzytelniające połączenie PPP (patrz plik */etc/ppp/chap-secrets* na Linuksie).

7.5. Debugowanie połączenia

Jeżeli wykonałeś wszystkie kroki z poprzednich punktów, połączenie powinno zadziałać od razu. W praktyce pewnie pojawią się jakieś komplikacje. W tym punkcie podam parę porad dotyczących tego, jak odnaleźć błąd.

W przypadku połączeń Windows-Linux (OpenSWAN) opartych na PSK błąd wystąpi prawdopodobnie gdzieś po stronie Linuksa. Konfiguracja Windowsa jest bowiem tak prosta, że trudno byłoby w niej coś źle zrobić.

Zacznij od przekierowania wszystkich logów systemowych do jednego pliku, aby łatwiej było podglądać na bieżąco, co się dzieje. W pisz do pliku */etc/syslog.conf* poniższą linijkę:

```
*.*                                     /var/log/all
```

Następnie przeładuj konfigurację demona Syslog. Wpisz komendę `killall -HUP syslogd`.

W produkcyjnym serwerze plik może szybko przyrastać. Pamiętaj, aby po zakończonych testach usunąć wpis z konfiguracji Sysloga.

Połącz się z serwerem na innej konsoli (jeśli pracujesz lokalnie, przełącz się na drugą konsolę). Wpisz polecenie:

```
tail -f /var/log/all
```

Na tej konsoli będziesz miał stały podgląd logów systemowych. Jeżeli na serwerze działają inne usługi, które możesz wyłączyć (np. poczta, dhcp itd.), zrób to — im mniej logów, tym łatwiej je przeglądać.

Przełącz się na pierwszą konsolę i sprawdź następujące rzeczy:

1. Czy działa demon L2TP — wpisz:

   ```
   ipsecgw:~# ps aux|grep l2tp
   root     16659  0.0  0.0   1656   560 ?        Ss   Oct31   0:00 xl2tpd
   ```

   Powinieneś zobaczyć proces. Jeżeli nie działa — uruchom go, wpisując `xl2tpd`. Sprawdź ponownie, czy widnieje na liście procesów. Jeżeli nie — zobacz, co mówią logi na drugiej konsoli.

2. Sprawdź poleceniem `netstat`, czy serwer nasłuchuje na portach 4500 (NAT Traversal), 500 (Pluto — IKE) oraz 1701 (L2TP). W tym celu wpisz polecenie:

   ```
   ipsecgw:~# netstat -anp|grep udp
   udp  0   0 127.0.0.1:4500           0.0.0.0:*         26771/pluto
   udp  0   0 85.98.29.251:4500        0.0.0.0:*         26771/pluto
   udp  0   0 192.168.10.98:4500        0.0.0.0:*         26771/pluto
   ```

```
udp  0   0 85.98.29.251:1701        0.0.0.0:*        16659/xl2tpd
udp  0   0 127.0.0.1:500            0.0.0.0:*        26771/pluto
udp  0   0 85.98.29.251:500         0.0.0.0:*        26771/pluto
udp  0   0 192.168.10.98:500         0.0.0.0:*        26771/pluto
```

Powinieneś zobaczyć wynik podobny do powyższego. W ostatniej kolumnie możesz zobaczyć nazwę procesu powiązanego z danym portem.

3. Upewnij się, czy firewall nie blokuje potrzebnych portów UDP oraz protokołu ESP. Najlepiej, jeśli na czas testów w ogóle wyłączysz firewalla, tzn. ustawisz domyślną politykę zapory na `ACCEPT`.
4. Sprawdź, czy w systemie na pewno zainstalowany jest program `pppd` — wpisz polecenie: `which pppd`.
5. Upewnij się, że w pliku *ipsec.conf* widnieje wpis `pfs=no`, który oznacza, że PFS nie jest konieczne (możliwe, gdy druga strona obsługuje). Implementacja Microsoftu nie obsługuje PFS, dlatego nie możemy go wymuszać.

Po stronie Windowsa debugowanie jest utrudnione z racji braku „sysloga”. Możesz zainstalować program Wireshark (następca Ethereala) — bardzo dobry sniffer sieciowy — i analizować nim fazy połączenia.

Jeżeli mimo sprawdzenia powyższych punktów dalej nie udaje się zestawić połączenia, spróbuj połączyć się, używając innego komputera klienckiego (innego Windowsa). Upewnij się, że żaden program typu firewall (zwłaszcza „kombajny” typu Norton Internet Security) nie blokuje połączenia.

7.6. Konfiguracja z uwierzytelnieniem przez certyfikaty

W niniejszym punkcie opiszę konfigurację podobną do poprzedniej, tzn. utworzymy bramę IPSec dla mobilnych użytkowników, z tą tylko różnicą, że do uwierzytelniania użyjemy certyfikatów X.509, a nie klucza współdzielonego. Konfiguracja taka jest zdecydowanie zalecana przy zdalnym dostępie pracowników, gdyż umożliwia w razie potrzeby unieważnienie certyfikatu użytkownikowi.

Przed przejściem do konfiguracji bramy IPSec oraz komputerów klienckich musimy przygotować klucze i certyfikaty dla serwera (bramy IPSec) oraz użytkowników. Szczegółowy opis generowania certyfikatów został omówiony w rozdziałach 3. (SSL) oraz 5. (OpenVPN). Aby nie powielać tych samych informacji, proszę Cię o zajrzenie do instrukcji zawartej w poprzednich rozdziałach.

Zakładam tutaj, że masz już wygenerowane klucze i certyfikaty dla serwera i użytkownika (na razie jeden użytkownik wystarczy).

Konfiguracja po stronie Linuksa (bramy VPN) znacząco się nie różni — więcej pracy będzie w systemie Windows.

7.6.1. Konfiguracja OpenSWAN z wykorzystaniem certyfikatów

Po stronie Linuksa — w stosunku do konfiguracji z użyciem PSK — zmianie ulegają tylko pliki *ipsec.conf* oraz *ipsec.secrets*, pozostałe konfiguracje są identyczne (demon L2TP, konfiguracja `pppd`).

Konfigurację wykonamy według następujących punktów:

1. Zapisz klucz prywatny serwera jako */etc/ipsec.d/private/serverkey.pem*.
2. Zapisz certyfikat serwera jako */etc/ipsec.d/certs/servercrt.pem*.
3. Zapisz certyfikat CA jako */etc/ipsec.d/cacerts/cacert.pem*.
4. Plik z listą unieważnionych certyfikatów (późniejszy etap) powinien znajdować się w katalogu */etc/ipsec.d/crls/*.
5. Dokonaj zmian w pliku */etc/ipsec.secrets*, tak aby miał następującą składnię:

   ```
   : RSA serverkey.pem "supertajnehaslo"
   ```

   gdzie:

   `serverkey.pem` to nazwa pliku z kluczem prywatnym, którego program OpenSWAN oczekuje w katalogu */etc/ipsec.d/private/*,

   `supertajnehaslo` to hasło do klucza prywatnego serwera. W przypadku gdy klucz prywatny nie jest zabezpieczony hasłem, można je pominąć. Wstawienie wartości `%prompt` spowoduje, że program OpenSWAN przy starcie będzie pytał o hasło do klucza.
6. Utwórz plik konfiguracyjny */etc/ipsec.conf* zgodny z listingiem 7.6.1.1.

Listing 7.6.1.1. *Konfiguracja programu OpenSWAN z wykorzystaniem certyfikatów*

```
version 2.0
config setup
        interfaces=%defaultroute
        plutodebug=none
        forwardcontrol=yes
        nat_traversal=yes
        virtual_private=%v4:10.0.0.0/8,%v4:172.16.0.0/12,
        ↪%v4:192.168.0.0/16,%v4:!192.168.10.0/24

conn roadwarrior-l2tp
        leftprotoport=17/1701
        rightprotoport=17/1701
        also=roadwarrior

conn roadwarrior
        auth=esp
        authby=rsasig
        compress=yes
        keyexchange=ike
        keyingtries=3
```

```
        pfs=no
        left=%defaultroute
        leftcert=/etc/ipsec.d/certs/servercrt.pem
        right=%any
        rightrsasigkey=%cert
        rightsubnet=vhost:%no,%priv
        rightca=%same
        auto=add

#Disable Opportunistic Encryption
include /etc/ipsec.d/examples/no_oe.conf
```

Opcja `rightrsasigkey=%cert` oznacza, że druga strona uwierzytelni się, przedstawiając swój certyfikat.

Opcja `rightca=%same` oznacza, że certyfikat drugiej strony (klienta) musi być wystawiony przez to samo CA co certyfikat serwera, czyli przez CA, którego certyfikat znajduje się na serwerze w pliku */etc/ipsec.d/cacerts/cacert.pem*.

7. Uruchom usługę IPSec. Wpisz polecenie: `ipsec setup start`.

W tym momencie brama jest już gotowa do działania. W następnym punkcie skonfigurujemy połączenie w systemie Windows.

7.7. Import certyfikatów w systemie Windows

Zakładam, że wygenerowałeś już użytkownikowi klucz prywatny i wystawiłeś mu certyfikat podpisany przez swoje CA. Powinieneś mieć już pliki *user.key* i *user.crt*. Będziesz musiał przekonwertować klucz i certyfikat do formatu P12 używanego w systemach Windows. W tym celu na komputerze CA wpisz polecenie:

```
ca:/etc/ssl# openssl pkcs12 -export -out user.p12 -inkey private/user.key -in user.crt
```

Program OpenSSL zapyta o hasło do klucza prywatnego, a następnie utworzy plik w formacie P12. Tak utworzony plik przegraj za pomocą bezpiecznego medium (SCP, pendrive) na komputer kliencki z systemem Windows. Przegraj też certyfikat CA — plik *CA.crt* (lub *cacert.pem* — w zależności od przyjętej konwencji).

Uwaga

> Nie instaluj i nie importuj certyfikatów w systemie Windows poprzez kliknięcie pliku. Ta metoda nie działa prawidłowo. Zamiast tego zawsze używaj przystawki MMC do importu kluczy i certyfikatów!

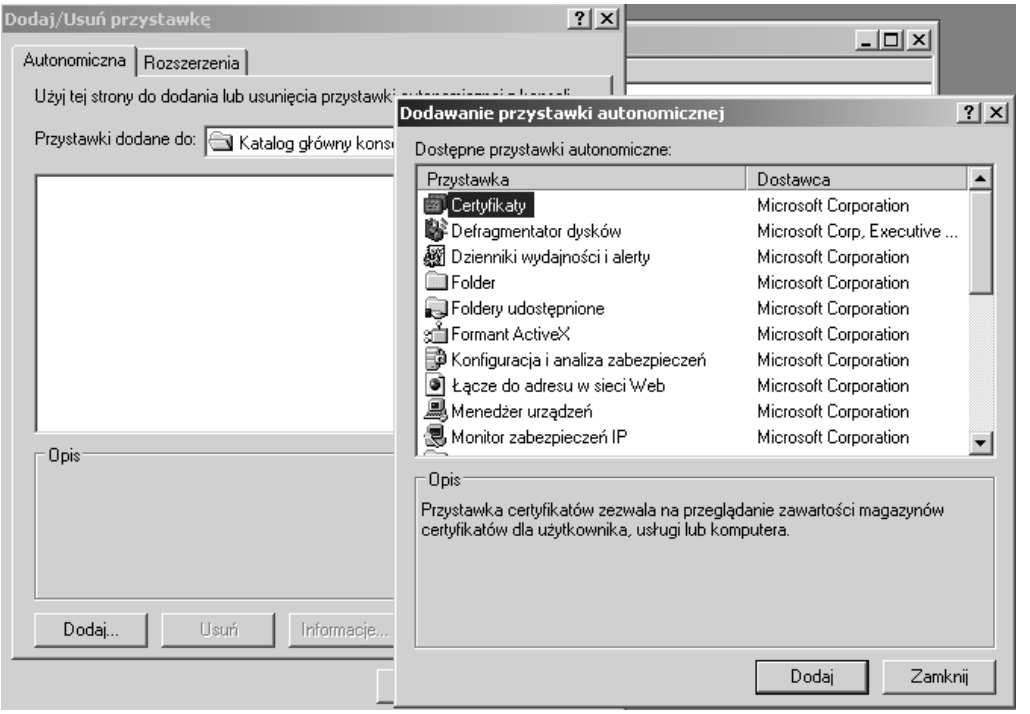
Procedurę importu certyfikatów przeprowadź według zamieszczonej instrukcji.

Utworzymy teraz przystawkę MMC, dzięki której będziesz mógł importować klucze i certyfikaty. Wpisz w pasku *Uruchom* menu *Start* polecenie `mmc` — uruchomi się konsola MMC.

Z menu *Plik* konsoli MMC wybierz opcję *Dodaj/Usuń przystawkę...* (patrz rysunek 7.7.1).

Rysunek 7.7.1. *Konsola MMC — wybierz opcję Dodaj przystawkę*

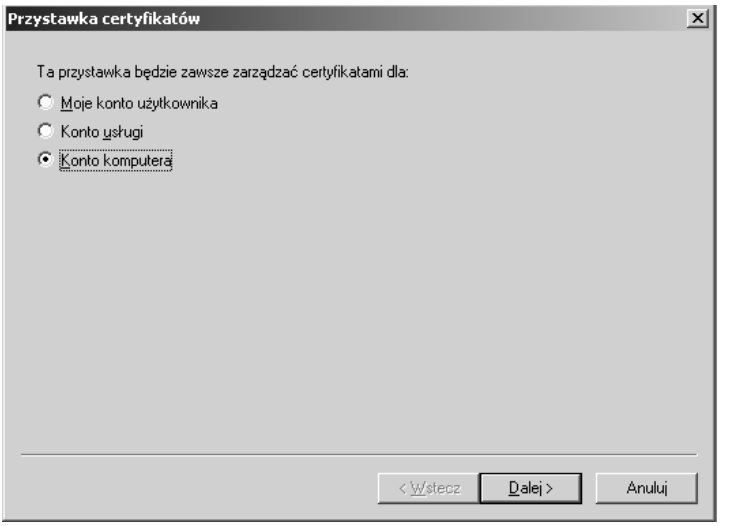
Pojawi się nowe okno *Dodaj/Usuń przystawkę* — kliknij przycisk *Dodaj* (patrz rysunek 7.7.2).



Rysunek 7.7.2. *Konsola MMC — dodawanie przystawki*

Na liście dostępnych przystawek zaznacz *Certyfikaty*, a następnie kliknij *Dodaj* — uruchomi się kreator konfiguracji przystawki. Wybierz opcję *Konto komputera*, następnie kliknij *Dalej* (patrz rysunek 7.7.3).

Rysunek 7.7.3. *Dodawanie przystawki Certyfikaty*

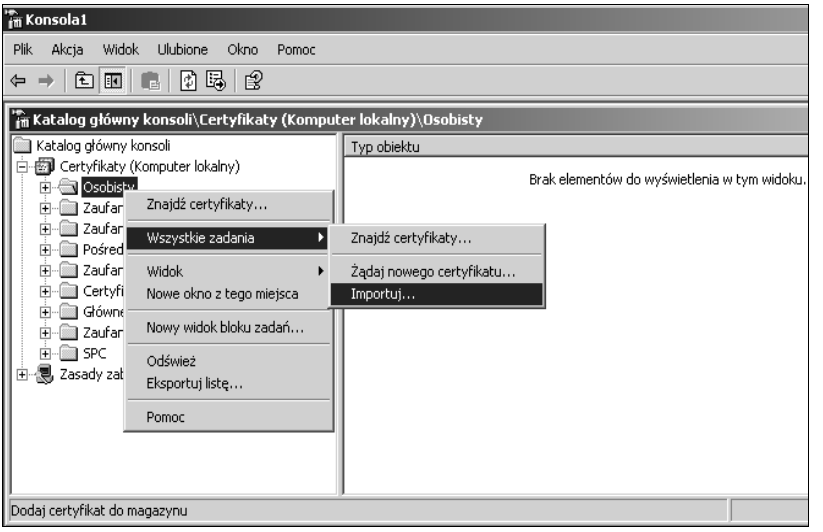


W następnym kroku wybierz opcję *Komputer lokalny* oraz kliknij przycisk *Zakończ*.

Z menu *Plik* wybierz opcję *Zapisz*, aby zapisać gotową przystawkę na dysku. Nazwij ją sobie, jak chcesz — np. `ipsec.mmc`.

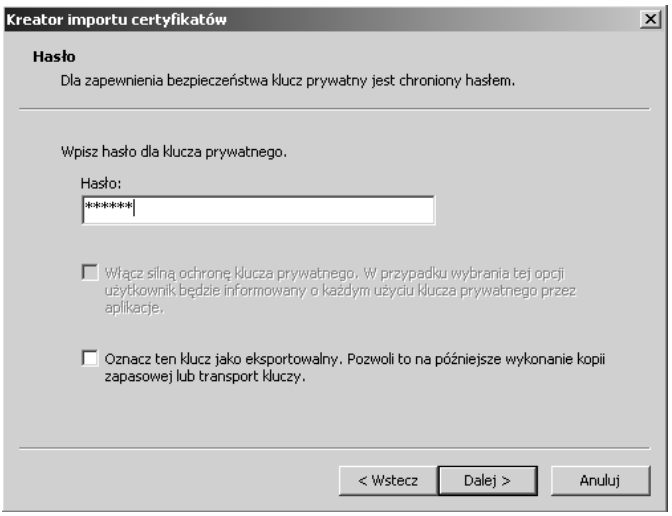
Mając gotową przystawkę, możesz zaimportować certyfikat. W tym celu rozwiń przystawkę *Certyfikaty*, a następnie kliknij prawym przyciskiem myszy folder *Osobisty*. Z menu wybierz *Wszystkie zadania*, a następnie opcję *Importuj...* — uruchomi się kreator importu certyfikatów (patrz rysunek 7.7.4).

Rysunek 7.7.4. *Z menu Wszystkie zadania wybierz opcję Importuj*



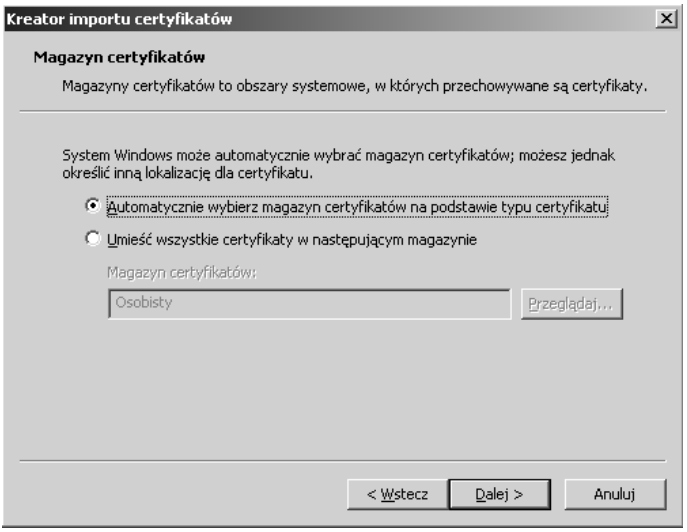
W kreatorze dodawania certyfikatów wskaż przygotowany wcześniej plik *user.p12*. Kreator zapyta o hasło do klucza prywatnego — podaj je (patrz rysunek 7.7.5).

Rysunek 7.7.5. *Import certyfikatu użytkownika — krok 1*



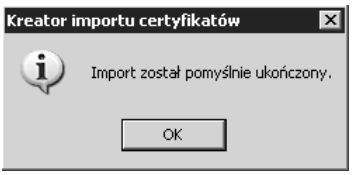
Kliknij przycisk *Dalej*, następnie wybierz opcję *Automatycznie wybierz magazyn certyfikatów na podstawie typu certyfikatu* (WAŻNE!) oraz ponownie kliknij przycisk *Dalej* (patrz rysunek 7.7.6).

Rysunek 7.7.6. *Import certyfikatu użytkownika — krok 2*

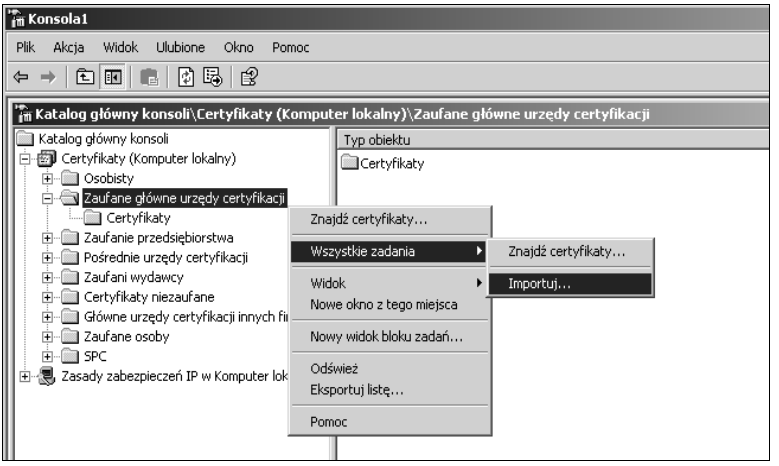


W kolejnym oknie kliknij przycisk *Zakończ*. Kreator powinien potwierdzić pomyślność importu certyfikatu (patrz rysunek 7.7.7).

Musisz jeszcze zainstalować certyfikat swojego CA w katalogu zaufanych urzędów certyfikacji. W tym celu kliknij prawym przyciskiem myszy katalog *Zaufane główne urzędy certyfikacji*, a następnie z menu wybierz opcję *Wszystkie zadania/Importuj* (patrz rysunek 7.7.8).

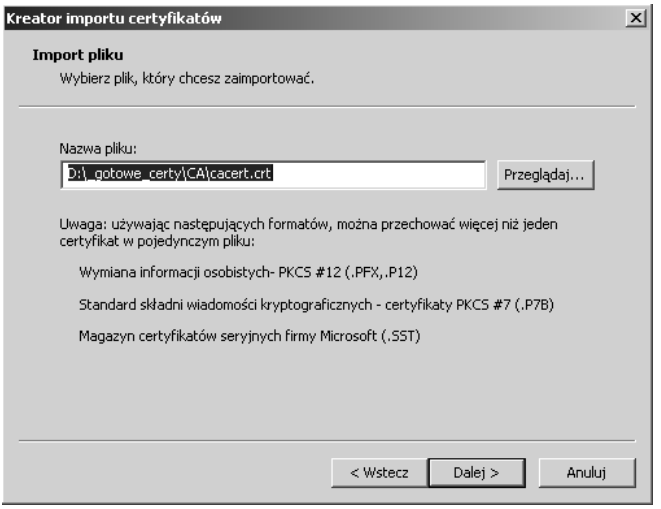


Rysunek 7.7.7. *Kreator powinien potwierdzić import*



Rysunek 7.7.8. *Wybierz opcję Importuj z menu*

Po wybraniu opcji *Importuj* uruchomi się kolejny kreator importu certyfikatów. Musisz wskazać plik z certyfikatem CA (*ca.crt*). Kreator importu certyfikatów oczekuje pliku z rozszerzeniem **.crt*, a nie **.pem*, dlatego przed importem musisz się upewnić, czy plik ma takie rozszerzenie (patrz rysunek 7.7.9).

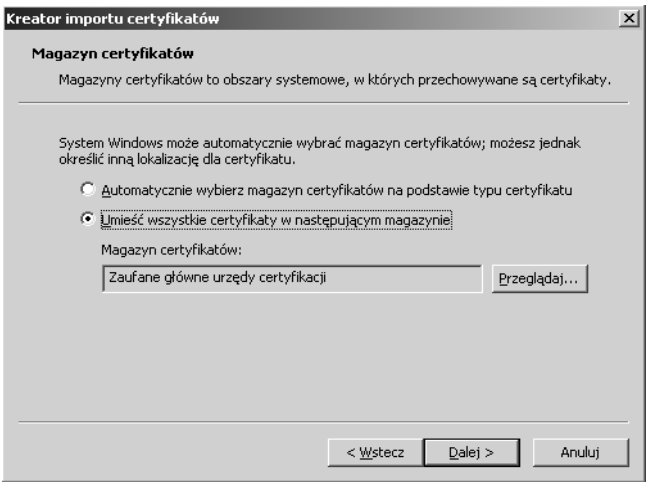


Rysunek 7.7.9. *Import certyfikatu CA — krok 1*

Kliknij przycisk *Dalej*, a następnie wybierz opcję *Umieść wszystkie certyfikaty w następującym magazynie — Zaufane główne urzędy certyfikacji* (patrz rysunek 7.7.10).

Wyjdź z konsoli, zapisując zmiany.

Rysunek 7.7.10.
Import certyfikatu CA — krok 2

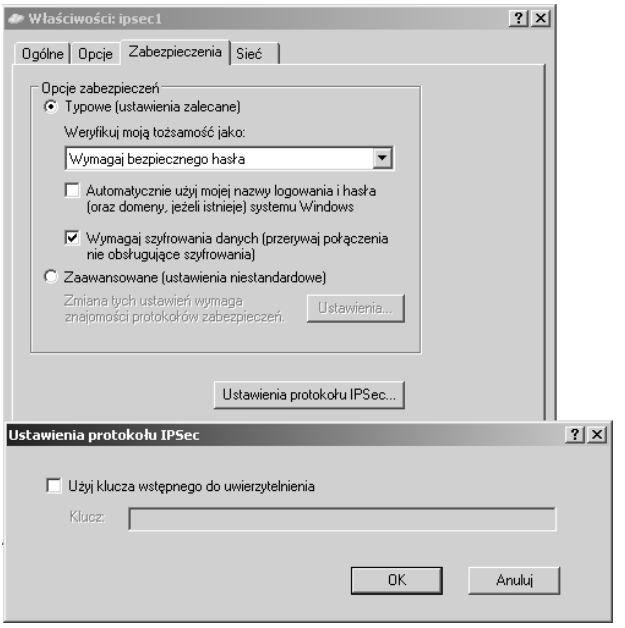


Jak widzisz, procedura importu jest trochę skomplikowana. Tak naprawdę złożona jest tylko za pierwszym razem. Mając przygotowaną przystawkę mmc (`ipsec.mmc`), możesz użyć jej na innych komputerach.

7.7.1. Konfiguracja połączenia

Konfigurację połączenia przeprowadź dokładnie w taki sam sposób, jak w przykładzie z kluczem współdzielonym. Jedyna różnica polega na tym, aby w zakładce *Zabezpieczenia* nie zaznaczać opcji *Ustawienia protokołu IPSec/Użyj klucza wstępnego do uwierzytelnienia* (patrz rysunek 7.7.1.1).

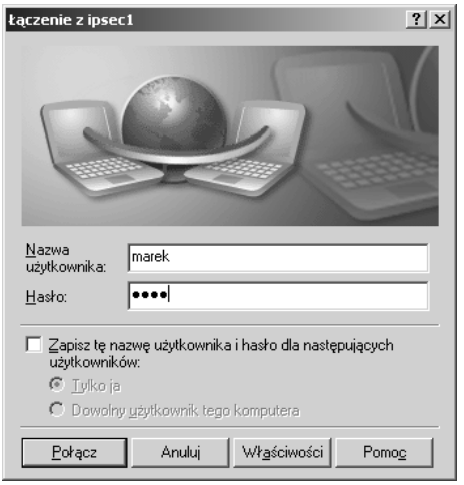
Rysunek 7.7.1.1.
Właściwości połączenia, zakładka Zabezpieczenia



Upewnij się, że w zakładce *Sieć* typ wirtualnej sieci prywatnej ustawiony jest na *Sieć VPN z protokołem L2TP IPSec*.

Zapisz zmiany i spróbuj się połączyć. W polu *Nazwa użytkownika* i *Hasło* podaj dane uwierzytelniające do połączenia PPP (plik */etc/ppp/chap-secrets* na serwerze) — patrz rysunek 7.7.1.2.

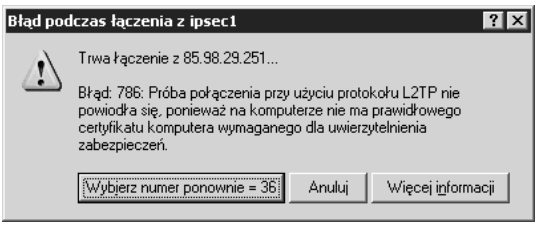
Rysunek 7.7.1.2.
Połączenie z bramą IPSec



Jeżeli nie wystąpią nieprzewidziane problemy, po chwili połączenie powinno zostać nawiązane.

W przypadku pojawienia się komunikatu o błędzie, mówiącego o braku certyfikatu wymaganego do uwierzytelnienia zabezpieczeń (patrz rysunek 7.7.1.3) upewnij się, czy zaimportowałeś poprawnie certyfikat CA w magazynie *Zaufane główne urzędy certyfikacji*. Najlepiej powtórz tę czynność jeszcze raz.

Rysunek 7.7.1.3.
Błąd połączenia IPSec — brak certyfikatu



W razie dalszych problemów z zestawieniem połączenia wykonaj kroki podane w punkcie 7.5.

Jeżeli połączenie zestawiło się, sprawdź po stronie systemu Windows, jaki adres IP otrzymał interfejs połączenia PPP/IPSec, oraz spróbuj „pingnąć” drugą stronę.

Na listingu 7.7.1.1 przedstawiono wycinek logów systemowych Linuksa. Zwróć uwagę na poszczególne fazy połączenia (wymiana kluczy przez IKE, nawiązanie SA, enkapsulacja PPP w L2TP). W logach widzimy też, że połączenie nadeszło z nieznanego adresu IP oraz że klient jest za NAT-em.

Listing 7.7.1.1. *Wycinek logów systemowych po stronie Linuksa*

```
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥responding to Main Mode from unknown peer 91.192.0.177
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥transition from state STATE_MAIN_R0 to state STATE_MAIN_R1
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥STATE_MAIN_R1: sent MR1, expecting MI2
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥NAT-Traversal: Result using draft-ietf-ipsec-nat-t-ike-02/03: peer is NATed
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥transition from state STATE_MAIN_R1 to state STATE_MAIN_R2
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥STATE_MAIN_R2: sent MR2, expecting MI3
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥Main mode peer ID is ID_DER_ASN1_DN: 'C=PL, ST=Slask, O=Helion, CN=user1'
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥I am sending my cert
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥transition from state STATE_MAIN_R2 to state STATE_MAIN_R3
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #8:
➥STATE_MAIN_R3: sent MR3, ISAKMP SA established {auth=OAKLEY_RSA_SIG
➥cipher=oakley_3des_cbc_192 prf=oakley_sha group=modp2048}
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #9:
➥responding to Quick Mode {msgid:e83ab5d8}
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #9:
➥transition from state STATE_QUICK_R0 to state STATE_QUICK_R1
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #9:
➥STATE_QUICK_R1: sent QR1, inbound IPsec SA installed, expecting QI2
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #9:
➥transition from state STATE_QUICK_R1 to state STATE_QUICK_R2
Nov  2 01:10:35 ipsecgw pluto[29172]: "roadwarrior-l2tp"[1] 91.192.0.177 #9:
➥STATE_QUICK_R2: IPsec SA established {ESP=>0xb66477ee <0xa49dc8ea xfrm=
➥3DES_0-HMAC_MD5 NATD=91.192.0.177:4500 DPD=none}


Nov  2 01:10:37 ipsecgw xl2tpd[29349]: Connection established to 91.192.0.177, 1701.
➥Local: 11685, Remote: 5 (ref=0/0).  LNS session is 'default'
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: start_pppd: I'm running:
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "/usr/sbin/pppd"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "passive"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "-detach"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "85.98.29.251:192.168.10.198"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "auth"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "require-chap"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "name"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "ipsec"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "file"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "/etc/ppp/options.l2tpd"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: "/dev/pts/1"
Nov  2 01:10:37 ipsecgw xl2tpd[29349]: Call established with 91.192.0.177,
➥Local: 27862, Remote: 1, Serial: 0
Nov  2 01:10:37 ipsecgw pppd[29356]: pppd 2.4.4 started by root, uid 0
Nov  2 01:10:37 ipsecgw pppd[29356]: Using interface ppp0
Nov  2 01:10:37 ipsecgw pppd[29356]: Connect: ppp0 <--> /dev/pts/1
Nov  2 01:10:39 ipsecgw pppd[29356]: found interface eth1 for proxy arp
Nov  2 01:10:39 ipsecgw pppd[29356]: local  IP address 85.98.29.251
Nov  2 01:10:39 ipsecgw pppd[29356]: remote IP address 192.168.10.198
```

W Linuksie informację o aktualnym stanie połączenia IPSec możesz uzyskać, wpisując polecenie:

```
ipsecgw:~# ip xfrm state
src 85.98.29.251 dst 91.192.0.177
proto esp spi 0xb66477ee reqid 16413 mode transport
replay-window 32
auth md5 0x4fc684eb720b08a2e0783b9a2dcbf31f
enc des3_ede 0xaee1a11e87622f31c06ddfbbd842a672575bc08bc2e29ca3
encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
src 91.192.0.177 dst 85.98.29.251
proto esp spi 0xa49dc8ea reqid 16413 mode transport
replay-window 32
auth md5 0x92d7ccce0c943d3742d9a3fd0304c656
enc des3_ede 0x9cc2b82f3c216d4947cd435fcec1f17ad3373bab5f381950
encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
```

Widzimy tutaj informacje o dwóch skojarzeniach SA (kierunek serwer-klient i klient-serwer). Identyfikatory SPI dla każdego SA zostały podkreślone. Widzimy także, że połączenie działa w trybie transportowym.

Na listingu 7.7.1.2 dla przykładu przedstawiłem komunikat zarejestrowany przez demon Syslog w przypadku próby nawiązania połączenia przez użytkownika legitymującego się złym certyfikatem (podpisanym przez inne CA — nie nasze).

Listing 7.7.1.2. *Komunikat o odrzuconym certyfikacie użytkownika*

```
Nov  2 02:21:44 ipsecgw pluto[29172]: "roadwarrior"[5] 91.192.0.182 #17:
↪issuer cacert not found
Nov  2 02:21:44 ipsecgw pluto[29172]: "roadwarrior"[5] 91.192.0.182 #17:
↪X.509 certificate rejected
Nov  2 02:21:44 ipsecgw pluto[29172]: "roadwarrior"[5] 91.192.0.182 #17:
↪no suitable connection for peer 'C=PL, ST=Rojca, O=rojcanet, OU=rojcanet,
↪CN=rojcanet_user1, E=marek@rojcanet.pl'
```

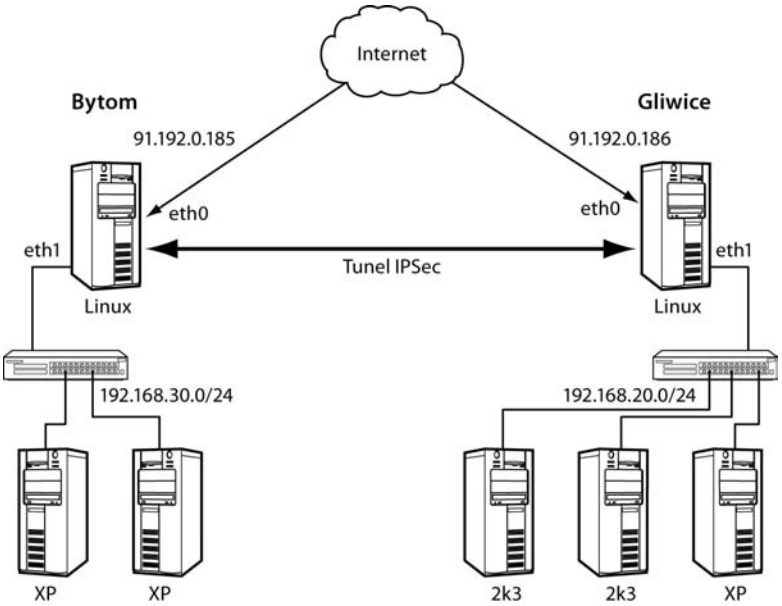
7.8. Dostęp z urządzeń PDA — Windows Mobile 2003, 2005, 2006

Z linuksowym serwerem VPN możesz się także połączyć, korzystając z urządzeń mobilnych typu palmtop. Zajrzyj do rozdziału 13., w którym opisałem procedurę importu certyfikatów SSL oraz konfigurację połączeń IPSec w systemach mobilnych.

7.9. Łączenie oddziałów firmy tunelem IPSec

W niniejszym punkcie stworzymy tunel łączący siedzibę firmy (Gliwice) z oddziałem (Bytom). W obu lokalizacjach routery działają pod kontrolą Linuksa z zainstalowanym programem OpenSWAN. Celem tunelu jest zapewnienie bezpiecznej komunikacji TCP/IP pomiędzy komputerami w oddziałach. Konfigurację przedstawia rysunek 7.9.1.

Rysunek 7.9.1. *Tunel IPSec/L2TP między oddziałami firmy*



Funkcjonalnie przykład jest podobny do rozwiązania z podrozdziału 5.4., różni się natomiast implementacją VPN. Zamiast działającego w warstwie użytkownika programu OpenVPN zestawimy tunel IPSec.

Do uwierzytelniania stron użyjemy tym razem kluczy RSA (klucz prywatny i publiczny). Jeżeli routery mają stałe IP, możesz użyć nawet klucza współdzielonego (hasło), ale pamiętaj, aby zablokować na firewallu możliwość łączenia się protokołem IPSec ze wszystkich hostów z wyjątkiem adresu IP „drugiej strony" (w przeciwnym razie ktoś będzie mógł próbować odgadnąć hasło, np. przez atak typu *brute force*).

Ponieważ implementacja IPSec w Linuksie nie wymaga użycia protokołu PPP oraz L2TP, konfiguracja jest tutaj znacznie prostsza niż w poprzednim rozdziale. W zasadzie gotowy plik konfiguracyjny znajduje się w przykładach dołączonych razem z pakietem OpenSWAN (*/etc/ipsec.d/examples/linux-linux.conf*).

Utwórz plik konfiguracyjny podany w listingu 7.9.1 — będzie to plik lokalizacji pierwszej (Gliwice).

Listing 7.9.1. *Konfiguracja IPSec Linux-Linux (lokalizacja Gliwice)*

```
version 2.0
config setup
        interfaces=%defaultroute
        forwardcontrol=yes
        rp_filter=0
        nat_traversal=no

conn linux-to-linux
        auth=esp
        authby=rsasig
        pfs=yes
        left=91.192.0.186
        leftsubnet=192.168.20.0/24
        leftrsasigkey=0sAQPuvae6KEw/yHijDjqHomCyLo8oO3H8wl3UExuTArCXtzc1DO5X2E8QFI
        ↪u0grLofzIzgoCy8AkoFthFPJIyDF3zKVH9ppMS8XQQL2naWp+YOm2cROstRlAfyvC/jF7GvW
        ↪lRIjxzHzCLCIJXihZmFZGNlku/DExLx5TjzqG/bXQ9DQ==
        right=91.192.0.185
        rightsubnet=192.168.30.0/24
        rightrsasigkey=0sAQODH/CRwexspJ6mu/bThfQzs84IpaHBYNs5MeDpxbiLdacZjM22PqOvb
        ↪VIqeQlYg4zHMAnB2EyUIgYHskJqyRmtmg6S5ELxnNHqvTE92KI5Bdicn458CowdqR2Jtc4tv
        ↪D7OWHv/RFzmt6WlkIHPiILAOkR2mSvATgI/QhZtNN4oaw==
        auto=start

include /etc/ipsec.d/examples/no_oe.conf
```

Oto znaczenie ważniejszych opcji:

`authby=rsasig` — uwierzytelnianie przez klucze RSA,

`left=91.192.0.186` — adres IP routera w Gliwicach („lewa strona" — lokalna dla routera),

`leftsubnet=192.168.20.0/24` — sieć LAN za routerem w Gliwicach,

`leftrsasigkey=0sAQPuvae6KE…` — klucz publiczny „lewej strony" (routera w Gliwicach),

`right=91.192.0.185` — adres IP bramy odległej lokalizacji (Bytom),

`rightsubnet=192.168.30.0/24` — sieć LAN za routerem w Bytomiu,

`rightrsasigkey=0sAQODH/CR…` — klucz publiczny prawej strony.

Zauważ, że jawnie umieściłem wpis `pfs=yes`. Nie musiałem tego robić, ponieważ jest to opcja domyślna. Moim zamiarem było podkreślenie, że w przypadku połączeń Linux-Linux PFS jest obsługiwana i nie należy jej wyłączać.

PFS (ang. *Perfect Forward Secrecy*) to, jak już wiemy, poufność doskonała. Przez pojęcie to rozumie się sposób wymiany kluczy sesyjnych w trakcie połączenia IPSec. Załączenie PFS zapewnia, że materiał klucza głównego może być używany do wygenerowania tylko jednego klucza sesji. Przed utworzeniem nowego klucza sesji jest przeprowadzana wymiana kluczy (algorytm Diffiego-Hellmana) w celu wygenerowania nowego materiału klucza głównego. Dzięki zastosowaniu PFS uzyskanie przez atakującego pojedynczego klucza pozwala mu na odczytanie tylko wiadomości zaszyfrowanych tym kluczem. Niestety, nie

wszystkie implementacje obsługują tę właściwość, dlatego w przykładach z użyciem klientów Windows umieszczaliśmy `pfs=no`.

Po stronie lokalizacji Bytom plik wygląda analogicznie — patrz listing 7.9.2.

Listing 7.9.2. *Konfiguracja IPSec Linux-Linux (lokalizacja Bytom)*

```
version 2.0
config setup
        interfaces=%defaultroute
        forwardcontrol=yes
        rp_filter=0
        nat_traversal=no

conn linux-to-linux
        auth=esp
        authby=rsasig
        pfs=yes
        left=91.192.0.185
        leftsubnet=192.168.30.0/24
        leftrsasigkey=0sAQODH/CRwexspJ6mu/bThfQzs84IpaHBYNs5MeDpxbiLdacZjM22PqOvbVI
        ↪qeQlYg4zHMAnB2EyUIgYHskJqyRmtmg6S5ELxnNHqvTE92KI5Bdicn458CowdqR2Jtc4tvD7OWH
        ↪v/RFzmt6WlkIHPiILAOkR2mSvATgI/QhZtNN4oaw==
        right=91.192.0.186
        rightsubnet=192.168.20.0/24
        rightrsasigkey=0sAQPuvae6KEw/yHijDjqHomCyLo8oO3H8wl3UExuTArCXtzc1DO5X2E8QFIu0gr
        ↪LofzIzgoCy8AkoFthFPJIyDF3zKVH9ppMS8XQQL2naWp+YOm2cROstRlAfyvC/jF7GvWlRI
        ↪jxzHzCLCIJXihZmFZGN1ku/DExLx5TjzqG/bXQ9DQ==
        auto=start

include /etc/ipsec.d/examples/no_oe.conf
```

Oczywiście z punktu widzenia routera w Bytomiu strona „lewa” to jego podsieć, a „prawa” — siedziba firmy w Gliwicach.

Musisz jeszcze wygenerować swoje klucze RSA i umieścić je w konfiguracjach po obu stronach. Zrób to według poniższych punktów:

1. Na obu routerach wpisz polecenie:

   ```
   gliwice:~#ipsec rsasigkey 1024 > /root/key.rsa
   ```

   Pliki będą miały postać podobną do podanej na listingu 7.9.3.

Listing 7.9.3. *Klucze RSA — publiczny i prywatny*

```
# RSA 1024 bits   hebaz   Wed Nov  7 13:40:56 2007
        # for signatures only, UNSAFE FOR ENCRYPTION

#pubkey=0sAQPCU0jPYnML3QzQzS8TsiyEJXj9p8uJZH1fgwLAjiahSA5NovxOOjfDKOJNtjOuTpxQ9bqrO
↪mu65FxrnLZAj31etSaaqyKGk3h6KEvIcPkESwuHpGBoxYsiuLJCmKGX5jZYkc3ckjnkqggL6eduC+4/FF
↪Gff5LzvwY5DSJLLGimLw==
        Modulus: 0xc25348cf62730bdd0cd0cd2f13b22c842578fda7cb89647d5f8302c08e26a148
        ↪0e4da2fc743a37c328e24db633ae4e9c50f5baab3a6bbae45c6b9cb6408f7d5eb5269aab2
        ↪28693787a284bc870f9044b0b87a46068c58b22b8b24298a197e6365891cddc9239e4aa08
        ↪0be9e76e0bee3f14519f7f92f3bf06390d224b2c68a62f
```

```
PublicExponent: 0x03
# everything after this point is secret
PrivateExponent: 0x206336cd3b132ca4d778223283485cc0b0e97f9bf74190bf8feb2b20
↪17b11ae157b79b2a135f094b317b0cf3b347b7c4b828f471df11f47b64bc9a1e6017ea3a2
↪88f02281ac42124d8af1381c2de4f15e13d70a1cb4b749abb6a6fbc72663b624305d89dd9
↪5ccbd5ace8df06ceff7c5dd60e01dc46a78603b68551f8ccc983c7
Prime1: 0xfdc0b4716ae73c2758d7834dc2837c06f55e5aacf3a678292422807926c1f0f99
↪099f8f1a0aa5694d4edda9a0697581e6c74c084456e108a79da28a056f53335
Prime2: 0xc40bd949170690740d3653701d3fadc0ceb8a5e8b15a5759301123b4cc7090ef3
↪5d4c137d962c71325a8d5248d77abeda388d3d1a5988a6578130dd608ba5c53
Exponent1: 0xa92b22f64744d2c4e5e50233d70252af4e3ee71df7c4501b6d6c55a619d6a0
↪a66066a5f66b1c39b88df3e71159ba3abef2f880582e49605c513c1b158f4e2223
Exponent2: 0x82b290db64af0af808cee24abe2a73d5df25c3f0763c3a3b7560c278884b0b
↪4a23e32b7a90ec84b76e708e185e4fc7f3c25b37e119105c43a56209395b26e837
Coefficient: 0x2552f68c894828dfbc997a9beb247d2b54439f26081f08dfc97f6caa1f11
↪170bbb7b3f7f740889e22efb727459f35f5e6c452f203381380587e2ed75f04174f8
```

2. Na obu routerach przekopiuj (myszką albo używając edytora vim) ciąg znaków zaczynający się od `#pubkey=…` i umieść go w pliku *ipsec.conf* przy parametrze `leftrsasigkey=`, przy czym usuń ze skopiowanego klucza publicznego ciąg znaków `#pubkey`. Dla przykładu parametr `leftrsasigkey=` z listingu 7.9.3 w pliku *ipsec.conf* powinien mieć wartość:

```
leftrsasigkey=0sAQPCU0jPYnML3QzQzS8TsiyEJXj9p8uJZH1fgwLAjiahSA5Novx0OjfDKOJNtj
↪OuTpxQ9bqrOmu65FxrnLZAj31etSaaqyKGk3h6KEvIcPkESwuHpGBoxYsiuLJCmKGX5jZYkc3ckj
↪nkqggL6eduC+4/FFGff5LzvwY5DSJLLGimLw==
```

   Możesz interesujący Cię ciąg wyciąć, używając programu awk:

```
cat /root/key.rsa |grep "#pubkey=" |awk -F'pubkey=' '{print $2}'
```

3. Przekopiuj ciąg klucza publicznego routera 1 (Gliwice) na router 2 (Bytom) i umieść go w pliku *ipsec.conf* przy parametrze `rightrsasigkey=`.

4. W analogiczny sposób klucz publiczny routera 2 (Bytom) umieść w pliku konfiguracyjnym routera 1 (Gliwice) przy parametrze `rightrsasigkey=`.

5. Otwórz do edycji plik */etc/ipsec.secrets* i umieść w nim następujący wpis:

```
91.192.0.186 91.192.0.185: RSA {
// część klucza prywatnego
// pobrana z pliku /root/key.rsa
}
```

   gdzie `91.192.0.186` to w powyższym przykładzie adres IP „lewej strony” (lokalny IP routera), a `91.192.0.185` to IP odległego routera (`right`). Na drugim routerze wpis wygląda odwrotnie.

   W sekcji pomiędzy nawiasami klamrowymi (`{ … }`) powinien znaleźć się klucz prywatny RSA.


Uwaga

> Z oczywistych względów kluczy prywatnych nie kompilujemy pomiędzy routerami. Klucz prywatny routera 1 (Gliwice) występuje tylko w pliku */etc/ipsec.secrets* routera w Gliwicach, analogicznie klucz prywatny routera 2 występuje tylko na routerze w Bytomiu.

6. Wstaw w sekcji pomiędzy nawiasami klamrowymi pliku */etc/ipsec.secrets* klucz prywatny. W tym celu przekopiuj z pliku */root/key.rsa* ciąg, począwszy od linii `Modulus:` do końca pliku. Przykładowy plik */etc/ipsec.secrets* przedstawiono na listingu 7.9.4.

Listing 7.9.4. *Przykładowy plik ipsec.secrets z kluczem prywatnym RSA*

```
91.192.0.186 91.192.0.185: RSA {
        Modulus: 0xeebda7ba284c3fc878a30e3a87a260b22e8f283b71fcc25dd4131b9302b097b
        ➥737350cee57d84f10148bb482b2e87f32338280b2f0092816d8453c92320c5df32951fda6
        ➥9312f174102f69da5a9f983a6d9c44eb2d46501fcaf0bf8c5ec6bd6951223c731f308b088
        ➥2578a166615918dd64bbf0c4c4bc794e3cea1bf6d743d0d
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent: 0x030f8f474f492ec4360f37fa2fb007cdc5851e0de3ec442206003eb6
        ➥65c72fe4cf7a1db44954d3109dcd473cc0a71552b1e4227872aac8b816770a9f69aad3417
        ➥39fab887852a159b8994f9e5f1836415d14746afc1aa71d72c683daa61e40b2fc693b5375
        ➥8b3b76701c2880e54d29296bdb5fbe1cda9013d4539b725f1fc2eb
        Prime1: 0xfee51c0c81a37ade390caf8a4399310b484d15cc299a401e24bc513e31999e190
        ➥dedf17535d0bd36c4598c8da0cb70247253239878b8766d975db1da80ab3ca1
        Prime2: 0xefc69e0568424d3f986c780f1da5df44c913b686338d2508adb841af8ff568c75
        ➥b24379018f3afbb9569a642e7c796ca87216f8708fb6b1c9af7930bf11b9ced
        Exponent1: 0xa9ee12b30117a73ed0b31fb1826620b2303363dd71118014187d8b7ecbbbbe
        ➥bb5e9ea0f8ce8b28cf2d91085e6b324ac2f6e217bafb25a4490f93cbe70072286b
        Exponent2: 0x9fd9beae458188d51048500a13c3ea2ddb6279aecd08c35b1e7ad6750aa39b
        ➥2f9218250abb4d1fd2639bc42c9a850f31af6b9faf5b524768674fb75d4b67bdf3
        Coefficient: 0xb271c7bf7ddca5f484b2f66d3e4b0cca0b61a28ae605d7f1db34d08a60a3
        ➥8c30df73cad9ea45eec5744fcfae202a3c4904d1a0fbd91ec36dfa11ddd2b16b7e90
        }
```

7. Uruchom tunel po obu stronach. Wpisz polecenie: `ipsec setup start`.

8. Sprawdź, czy połączenie IPSec zostało zestawione. Wpisz polecenie: `ip xfrm state`.