Bezpieczeństwo systemów informatycznych

ĆWICZENIE VPN

1. Tunele wirtualne

1.1 Narzędzie OpenVPN

OpenVPN jest narzędziem służącym do tworzenia szyfrowanego połączenia VPN w sieci TCP/IP. Instalacja nie wymaga ingerencji w jądro systemu operacyjnego Linux przez co rozwiązanie to jest proste i łatwe w uruchomieniu. Tunel jest tworzony z wykorzystaniem wirtualnych interfejsów sieciowych TUN/TAP. Utworzenie tunelu sprowadza się do utworzenia pliku konfiguracyjnego opisującego parametry połączenia i uruchomienie programu openvpn z tym plikiem jako parametr. Z drugiej strony tunelu również wykorzystujemy ten sam program ale w roli klienta i przyłączamy się do poprzednio uruchomionego serwera. Bardzo dużą zaletą programu OpenVPN jest możliwość tworzenia szyfrowanych tuneli z hostami korzystającymi z systemu Windows ,co w niektórych sytuacjach może okazać się kluczowe przy wyborze darmowego oprogramowania do tworzenia sieci VPN.

OpenVPN pozwala tworzyć szyfrowane połączenia w oparciu o dwie metody:

1. tzw „pre-shared key" czyli sytuacje gdy dwie strony współdzielą między sobą tajny klucz służący do uwierzytelnia stron i szyfrowania komunikacji. OpenVPN wykorzystuje algorytm blowfish z 128 bitowym kluczom. (Static key mode)

Wygenerowanie współdzielonego klucza następuje po wydaniu komendy:

```
% openvpn --genkey --secret shared.key
```

2. wykorzystanie modułu OpenSSL dla kryptografii klucza publicznego czyli popularne certyfikaty SSL (SSL/TLS mode)

Przykład uruchomienia programu openvpn w roli serwera przyjmującego połączenia:

```
% openvpn --config /etc/openvpn/static.conf
```

Opcja w pliku konfiguracyjnym decyduje, która strona połączenia jest serwerem, a która klientem.

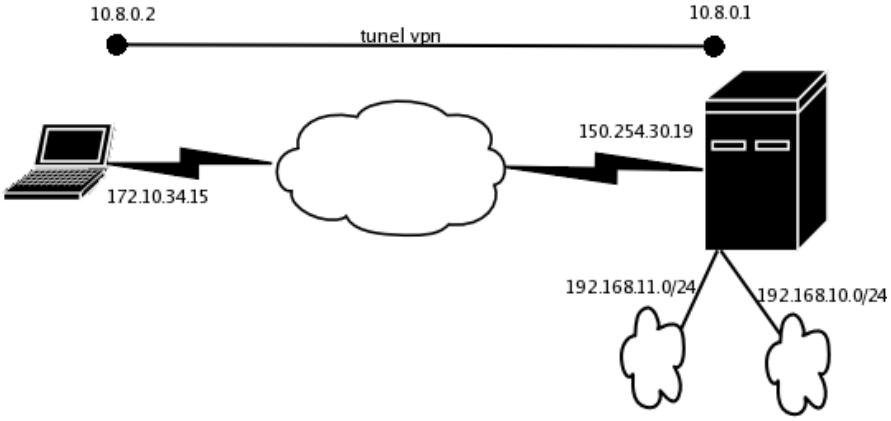
Przykład uruchomienia programu openvpn w roli klienta nawiązującego połączenie z serwerem:

```
% openvpn --config /etc/openvpn/client.conf
```

Oczywiście istnieje możliwość podania innych opcji w momencie uruchomienia programu openvpn ale w większości przypadków o wiele wygodniejsze jest przygotowanie odpowiedniego pliku konfiguracyjnego dla strony serwera i klienta.

1.2 Przykładowa architektura sieci:

Koniec tunelu od strony serwera ma adres IP: 10.8.0.1, a od strony klienta 10.8.0.2 (patrz opcja ifconfig w pliku konfiguracyjnym dla serwera: p.1.3.1 i klienta: p.1.3.2)



1.3 Uwierzytelnianie i szyfrowanie następuje przez mechanizm pre-shared keys.

1.3.1 Plik konfiguracyjny dla serwera

Serwer jest dostępny pod adresem IP: 150.254.30.19 i nasłuchuje na porcie 1194.

```
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret /root/openvpn/shared.key
proto tcp-server
daemon
verb 4
log-append /var/log/openvpn.log
keepalive 10 900
inactive 3600
comp-lzo
```

`verb` – oznacza poziom szczegółowości powiadomień o działaniu openvpn

Pozostałe opcje konfiguracyjne opisane są w pkt 1.3.2.

1.3.2 Przykład pliku konfiguracyjnego dla klienta łączącego się z serwerem.

Uwierzytelnianie i szyfrowanie następuje przez mechanizm pre-shared keys.

```
# remote <ip serwera> <port, na którym nasłuchuje serwer>
remote 150.254.30.19 1194
#
# urządzenie (zostawić bez zmian)
dev tun
proto tcp-client
# ip dla tunelu między punktami
ifconfig 10.8.0.2 10.8.0.1
#
# nazwa pliku z wygenerowanym kluczem
secret shared.key
#
keepalive 10 60
# trasa do sieci wewnętrznej za bramką VPN (po stronie serwera)
# route <adres sieci> <maska sieci>
route 192.168.10.0 255.255.255.0
route 192.168.11.0 255.255.255.0
#
comp-lzo
```

Opisy użytych opcji w powyższym przykładzie znajdują się w podręczniku systemowym dla programu openvpn (`man openvpn`). Dla przykładu:

`remote` – oznacza jaki adres IP ma serwer i na jakim porcie nasłuchuje serwer
`dev` – oznacza, jaki sterownik jest wykorzystany do utworzenia połączenia VPN
`proto` – oznacza jaki protokół będzie wykorzystywany do przenoszenia zaszyfrowanego ruchu. Za pomocą tego słowa kluczowego ustanawia się kto w połączeniu jest serwerem, a kto klientem.
`ifconfig` – polecenie wykorzystane w pliku konfiguracyjnym do nadania adresów IP obu końcom tunelu. Tworzone jest połączenie punkt-punkt.
`keepalive` – polecenie pomocnicze ułatwiające utrzymywanie tunelu i sprawdzanie dostępności klienta.
`route` – polecenie pozwala podać adresy sieci jakie mają być dostępne przez połączenie VPN
`comp-lzo` – polecenie uaktywniające kompresję przesyłanych danych

Serwer udostępnia dwie sieci: 192.168.10.0/24 i 192.168.11.0/24 (patrz plik konfiguracyjny)

1.4 Uwierzytelnianie stron i szyfrowanie ruchu sieciowego za pomocą certyfikatów SSL

1.4.1 Przykład pliku konfiguracyjnego dla serwera.

```
log-append /var/log/openvpn-server.log
status /var/log/openvpn-server-status.log
proto tcp-server
daemon
comp-lzo
mlock
persist-tun
persist-local-ip
persist-remote-ip
persist-key
##########SSL##########################
tls-server
tls-remote polanka
ca cacert.pem
dh dh1024.pem
cert newcert.pem
key newreq.pem
cipher aes-256-cbc
######################################
dev tun
ifconfig 10.8.0.1 10.8.0.2
verb 4
keepalive 5 240
max-clients 1
inactive 3600
```

`mlock` – polecenie zwiększające poziom bezpieczeństwa programu openvpn, wymuszające nie zapisywanie kluczy na dysk twardy.
`persist-local-ip` – pomocnicze polecenie pozwalające zapamiętać lokalny adres IP
`persist-remote-ip` – podobnie jak wyżej ale dotyczy adresu zdalnego hosta
`user`, `group` – polecenia wymuszające zrzucenie uprawnień do poziomu użytkownika nieuprzywilejowanego
`tls-server` – polecenie pozwalające określić stronę serwera podczas negocjacji SSL
`tls-remote` – polecenie pozwala podać nazwę klienta, który może nawiązać połączenie z serwerem (podana nazwa musi być zgodna z *common name* w certyfikacie tego klienta)
`cipher` – pozwala podać jaki algorytm ma zostać użyty do szyfrowania przesyłanych danych

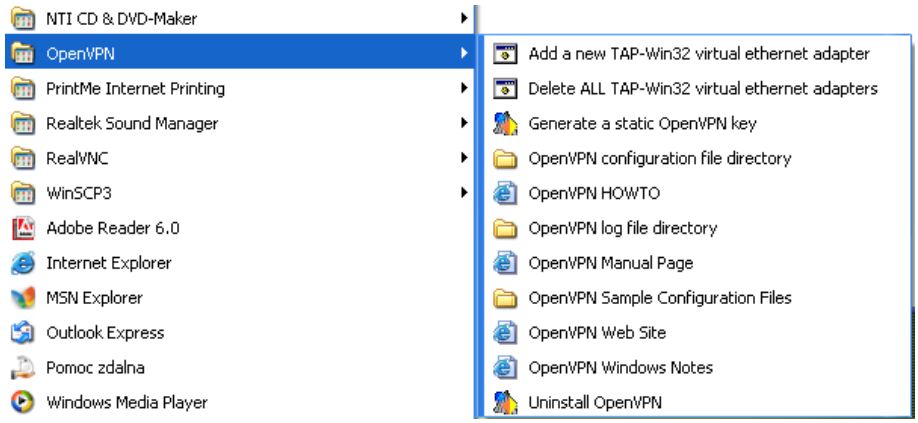
1.4.2 Przykład pliku konfiguracyjnego dla klienta.

```
log-append /var/log/openvpn-client.log
proto tcp-client
verb 4
comp-lzo
mlock
persist-tun
persist-local-ip
persist-remote-ip
persist-key
#######SSL#####################################
tls-client
tls-remote cpn
ca cacert.pem
cert newcert.pem
key newreq.pem
cipher aes-256-cbc
##############################################
# remote <ip serwera> <port, na którym nasłuchuje serwer>
# urządzenie (zostawić bez zmian)
dev tun
# ip dla tunelu między punktami
ifconfig 10.8.0.2 10.8.0.1
# trasa do sieci wewnętrznej za bramką VPN (po stronie serwera)
# route <adres sieci> <maska sieci>
route 192.168.10.0 255.255.255.0
route 192.168.11.0 255.255.255.0
```

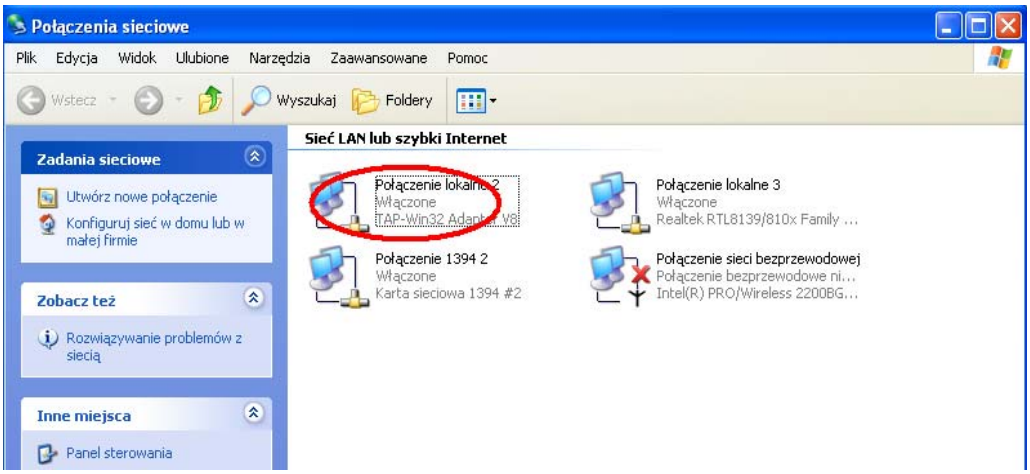
1.5 Tworzenie tuneli między systemami Linux a Windows.

OpenVPN jest narzędziem, które pozwala na tworzenie tuneli również pod Windows. Program openvpn posiada specjalną wersję pod Windows, która pozwala na uruchomienie tunelu np. między Linuksem a Windows. Poniżej pokazano jak można utworzyć taki tunel VPN.

Po zainstalowaniu OpenVPN pojawia się w menu START:



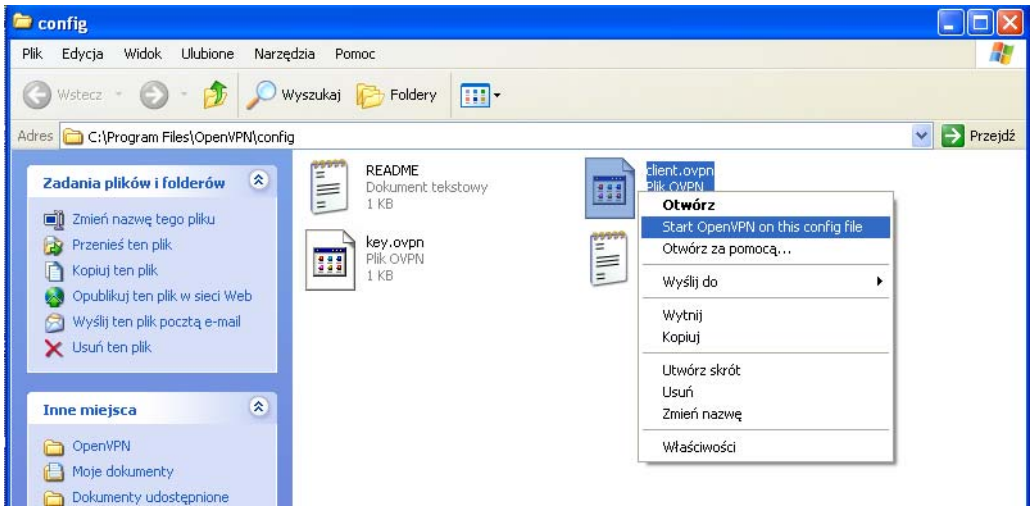
Po zainstalowaniu otrzymujemy również nowe urządzenie sieciowe TAP-win32 Adapter przez które będzie można nawiązać szyfrowane połączenie z odległym hostem.



1.5.1 Uruchomienie openvpn pod Windows.

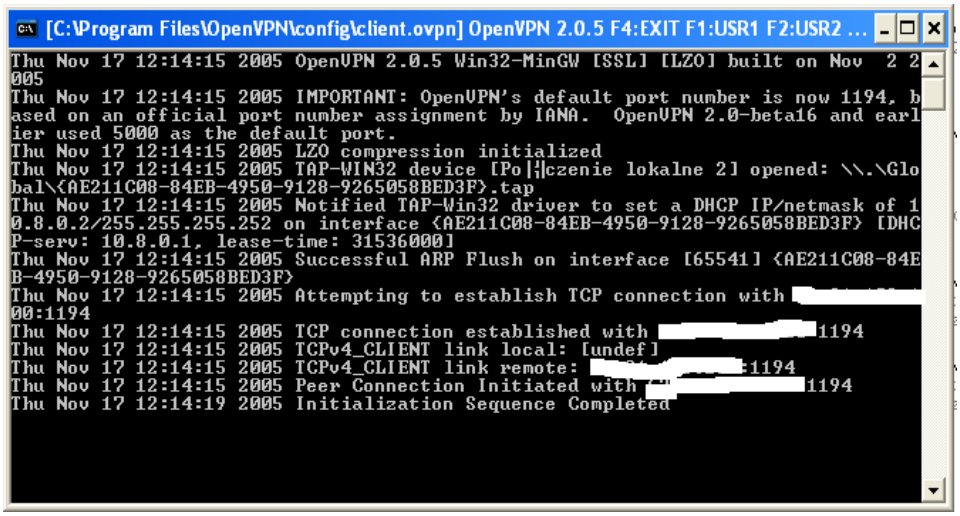
Wystarczy wybrać z menu kontekstowego odpowiednią opcję.
Jest kilka rzeczy na które trzeba zwrócić uwagę:

- plik konfiguracyjny musi mieć rozszerzenie .ovpn
- plik konfiguracyjny musi znajdować się w katalogu config programu openvpn (patrz ramka z adresem folderu)

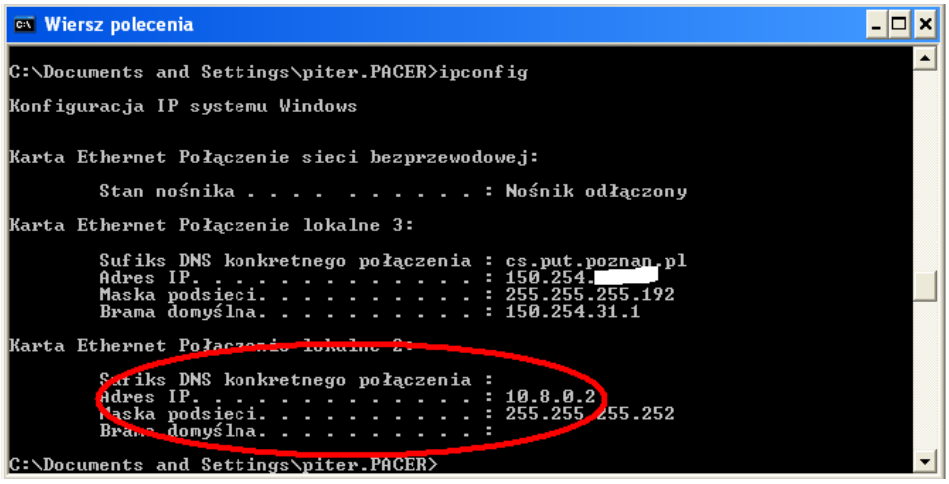


Po uruchomieniu pokazuje nam się okno konsoli tekstowej z logami programu openvpn.

Jeżeli zdecydujemy się aby logowanie odbywało się do pliku na konsoli nie zobaczymy logów. Utworzenie tunelu następuje po zakończeniu procedury negocjacji połączenia (patrz ostatnia linia na poniższym zrzucie ekranu).



Nowy adres IP po ustanowieniu tunelu:



Literatura

strona domowa projektu OpenVPN: http://openvpn.net

Zadania:

1. Skonfiguruj połączenie VPN przy użyciu mechanizmu pre-shared key.
2. Skonfiguruj połączenie VPN przy użyciu certyfikatów SSL.
3. Skonfiguruj połączenie VPN w sytuacji gdy jeden host będący końcem tunelu działa pod kontrolą systemu Windows.

Dodatek

Generacja certyfikatów SSL

Wykorzystany zostanie skrypt CA.sh z katalogu /etc/ssl/misc

1. utworzenie certyfikatu CA(centrum certyfikacji):

```
./CA.sh -newca
```

Należy podać wszystkie dane potrzebne do utworzenia certyfikatu.

2. utworzenie pliku z kluczem prywatnym dla danej strony tunelu wraz z prośbą o podpisanie klucza:

```
./CA.sh -newreq
```

Należy podać wszystkie dane potrzebne do utworzenia certyfikatu dla danej strony połączenia.

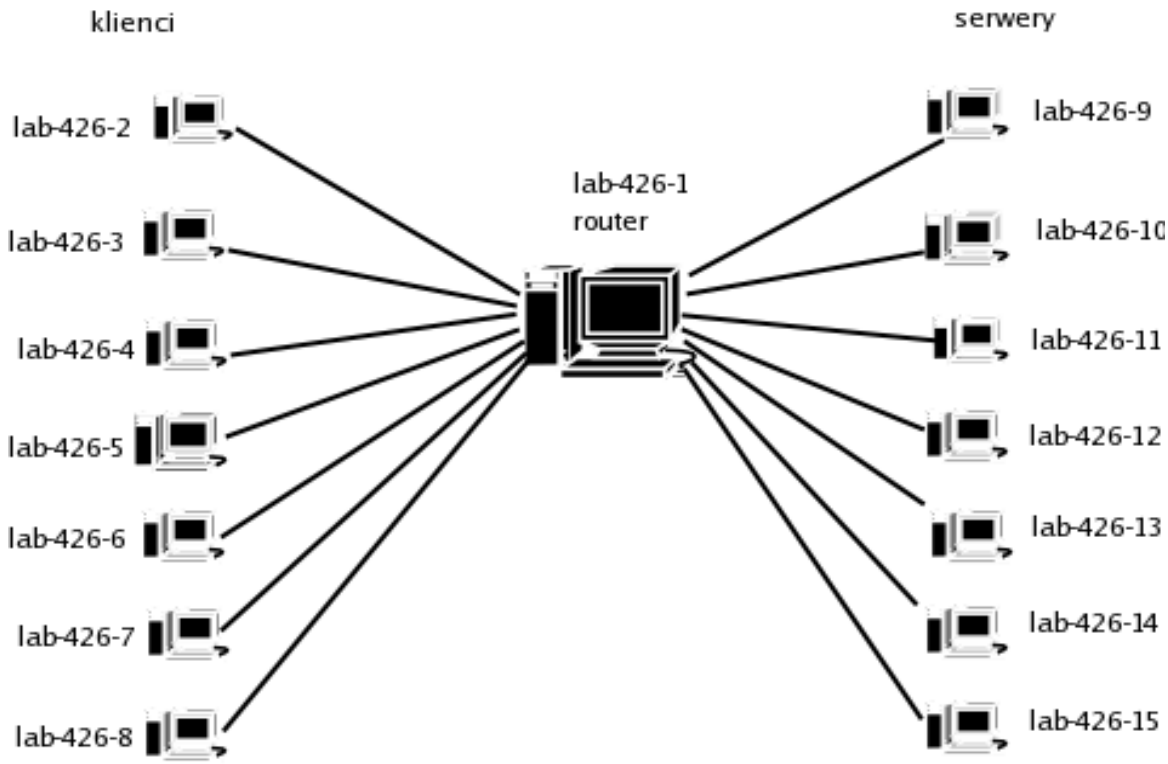
3. podpisanie prośby o certyfikat z pkt. 2:

```
./CA.sh -sign
```

Po wykonaniu wszystkich kroków będziemy dysponować trzema plikami:

cacert.pem – certyfikat dla CA
newreq.pem – klucz prywatny dla danej strony połączenia
newcert.pem – certyfikat z kluczem publicznym dla danej strony połączenia

Topologia środowiska na potrzeby ćwiczeń z sieci VPN:



Adresacja na potrzeby wykonania ćwiczeń:

eth1	*host*	*Eth0:1*	*Eth0:x*	*router*	*Eth1:x*	*Eth0:1*	*host*	*eth1*
172.16.2.2	Lab-426-2	192.168.2.2	192.168.2.1	Lab-426-1	192.168.9.1	192.168.9.2	Lab-426-9	172.16.9.2
172.16.3.2	Lab-426-3	192.168.3.2	192.168.3.1	Lab-426-1	192.168.10.1	192.168.10.2	Lab-426-10	172.16.10.2
172.16.4.2	Lab-426-4	192.168.4.2	192.168.4.1	Lab-426-1	192.168.11.1	192.168.11.2	Lab-426-11	172.16.11.2
172.16.5.2	Lab-426-5	192.168.5.2	192.168.5.1	Lab-426-1	192.168.12.1	192.168.12.2	Lab-426-12	172.16.12.2
172.16.6.2	Lab-426-6	192.168.6.2	192.168.6.1	Lab-426-1	192.168.13.1	192.168.13.2	Lab-426-13	172.16.13.2
172.16.7.2	Lab-426-7	192.168.7.2	192.168.7.1	Lab-426-1	192.168.14.1	192.168.14.2	Lab-426-14	172.16.14.2
172.16.8.2	Lab-426-8	192.168.8.2	192.168.8.1	Lab-426-1	192.168.15.1	192.168.15.2	Lab-426-15	172.16.15.2

Adresacja tuneli:

klienci(10.8.0.2) $\Longleftrightarrow$ serwery(10.8.0.1)