Bezpieczeństwo systemów informatycznych

ĆWICZENIE VPN TLS

1. Tunele wirtualne VPN TLS

1.1 OpenVPN

OpenVPN jest narzędziem służącym do tworzenia tuneli VPN przy wykorzystaniu protokołu TLS, przez co nie wymaga ingerencji w jądro systemu operacyjnego. Tunel jest tworzony z wykorzystaniem wirtualnych interfejsów sieciowych TUN/TAP.

OpenVPN pozwala uwierzytelniać strony tunelu poprzez dwie metody:

1. *static key mode*: strony współdzielą tajny wstępny klucz (in. *pre-shared key*).

   Wygenerowanie współdzielonego klucza następuje komendą:

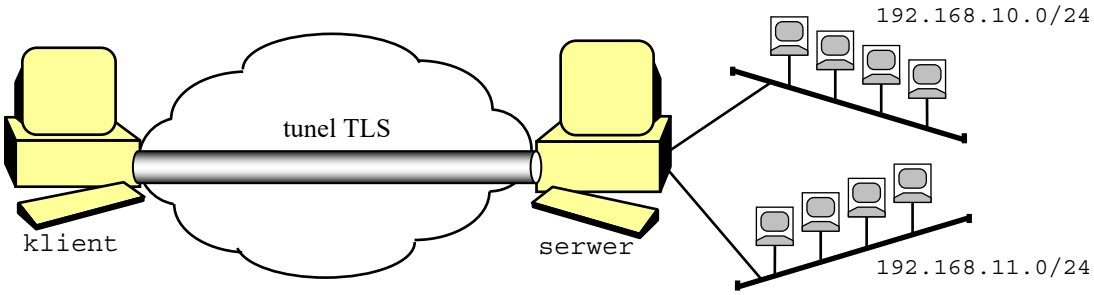
```
openvpn --genkey --secret shared.key
```

2. *TLS mode*: certyfikaty X.509 i wykorzystanie infrastruktury klucza publicznego PKI.

Utworzenie tunelu sprowadza się do utworzenia pliku konfiguracyjnego opisującego parametry tunelu i uruchomienie programu openvpn ze wskazaniem tej konfiguracji.

1.2 Przykładowy tunel VPN

W odróżnieniu od IPsec, w tunelu utworzonym przez OpenVPN wyróżnia się role serwera i klienta. Zawartość pliku konfiguracyjnego decyduje, która strona połączenia jest serwerem, a która klientem. Koniec tunelu (wirtualny interfejs tunelowy w systemie operacyjnym) od strony serwera ma adres IP: 10.8.0.1, a od strony klienta 10.8.0.2 (patrz opcja `ifconfig` w plikach konfiguracyjnych poniżej). Serwer jest dostępny pod adresem IP: 150.254.32.19 i nasłuchuje na porcie 1194. Serwer udostępnia klientowi dwie sieci 192.168.10.0/24 i 192.168.11.0/24 znajdujące się za tunelem.



Przykładowy plik konfiguracyjny dla serwera (`/etc/openvpn/static.conf`):

```
proto tcp-server
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret /root/openvpn/shared.key
keepalive 10 900
inactive 3600
cipher aes-256-cbc
```

Uruchomienie serwera:

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

Przykład pliku konfiguracyjnego dla klienta (`/etc/openvpn/client.conf`):

```
proto tcp-client
# remote <ip_serwera> <port_serwera>
remote 150.254.32.19 1194
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret /root/openvpn/shared.key
keepalive 10 60
cipher aes-256-cbc
# trasa do sieci wewnętrznej za bramką VPN (po stronie serwera)
route 192.168.10.0 255.255.255.0
route 192.168.11.0 255.255.255.0
```

Uruchomienie klienta:

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

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

`remote` – określa jaki adres IP ma serwer i na jakim porcie nasłuchuje serwer
`dev` – określa jaki sterownik jest wykorzystany do utworzenia połączenia VPN
`ifconfig` – polecenie nadania adresów IP obu końcom tunelu
`keepalive` – polecenie pomocnicze ułatwiające utrzymywanie tunelu i sprawdzanie dostępności klienta.
`cipher` – pozwala podać jaki algorytm ma zostać użyty do szyfrowania przesyłanych danych
`route` – polecenie pozwala podać adresy sieci jakie mają być dostępne przez połączenie VPN
`verb` – określa poziom szczegółowości powiadomień o działaniu openvpn (domyślnie na konsoli, chyba że zdecydujemy się na rejestrowanie w pliku logu)

1.3 Uwierzytelnianie stron i szyfrowanie ruchu sieciowego z wykorzystaniem certyfikatów

Przykład pliku konfiguracyjnego dla serwera:

```
proto tcp-server
dev tun
ifconfig 10.8.0.1 10.8.0.2
##########TLS##########################
tls-server
verify-x509-name lab-sec-44 name
ca cacert.pem
dh dh.pem
cert newcert.pem
key newkey.pem
######################################
persist-tun
persist-local-ip
persist-remote-ip
persist-key
mlock
keepalive 5 240
max-clients 1
inactive 3600
```

`verify-x509-name` – polecenie pozwala podać nazwę *Common Name* drugiej strony (podana nazwa musi być zgodna z *Common Name* w certyfikacie)
`persist-local-ip` – pomocnicze polecenie pozwalające wykorzystać ten sam lokalny adres IP również w trakcie kolejnych sesji
`persist-remote-ip` – podobnie jak wyżej, ale dotyczy adresu zdalnego
`mlock` – zabrania ewentualnego zrzutu newralgicznych danych (kluczy) na dysk twardy w trakcie stronicowania pamięci wirtualnej

Przykład pliku konfiguracyjnego dla klienta:

```
proto tcp-client
remote 150.254.32.19 1194
dev tun
ifconfig 10.8.0.2 10.8.0.1
#######TLS#####################################
tls-client
verify-x509-name lab-sec-server name
ca cacert.pem
cert newcert.pem
key newkey.pem
################################################
persist-tun
persist-local-ip
persist-remote-ip
persist-key
mlock
# trasa do sieci wewnętrznej za bramką VPN (po stronie serwera)
route 192.168.10.0 255.255.255.0
route 192.168.11.0 255.255.255.0
```

1.4 Konfiguracja serwera dla wielu klientów

Przykład pliku konfiguracyjnego dla serwera:

```
proto tcp-server
dev tun
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt  # opcjonalne
max-clients 10
##########TLS##########################
tls-server
ca cacert.pem
dh dh.pem
cert newcert.pem
key newkey.pem
#######################################
persist-tun
persist-local-ip
persist-remote-ip
persist-key
mlock
keepalive 5 240
inactive 3600
```

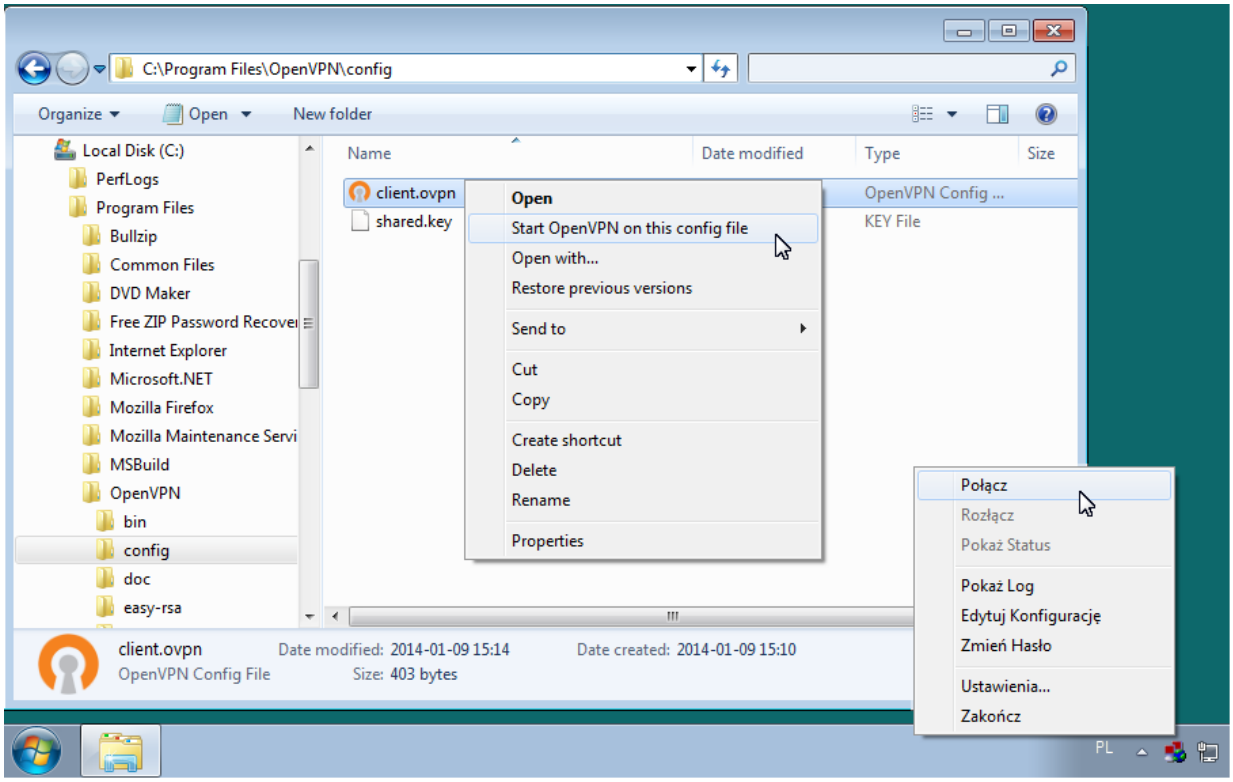
2. Tworzenie tuneli VPN pomiędzy systemami Linux a Windows

OpenVPN pozwala na tworzenie tuneli VPN również w systemie operacyjnym Windows oraz zestawienie tunelu np. między Windows a Linuksem. Poniżej pokazano jak można utworzyć taki tunel.

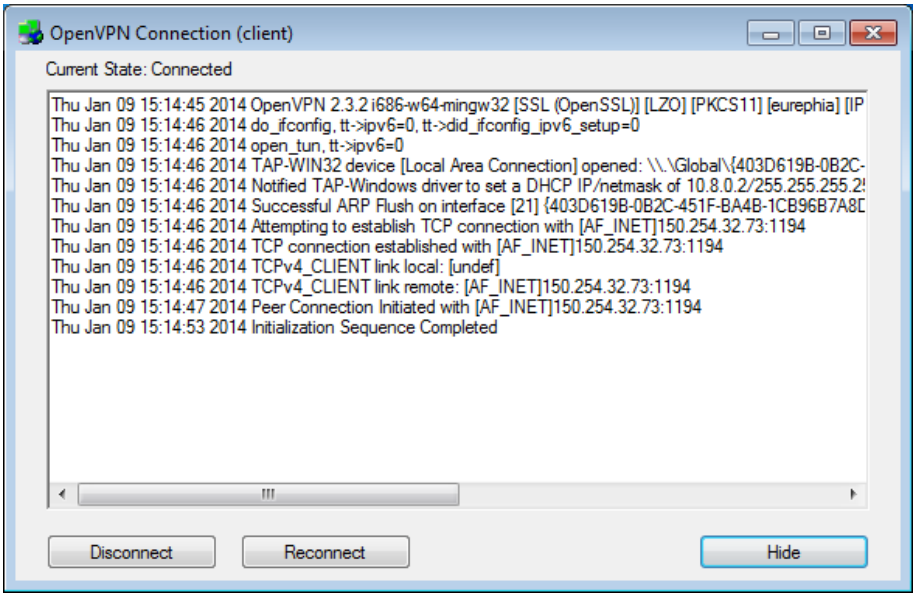
Po zainstalowaniu OpenVPN otrzymujemy nowy interfejs sieciowy TAP, przez który realizowane będą szyfrowane połączenia VPN:

Jest kilka rzeczy na które trzeba zwrócić uwagę:
– plik konfiguracyjny musi mieć rozszerzenie .ovpn
– i musi znajdować się w katalogu config programu openvpn.

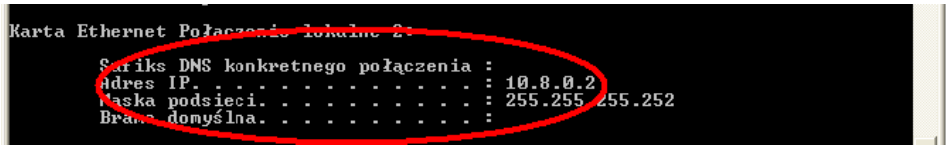
W celu uruchomienia skonfigurowanego tunelu wystarczy wybrać odpowiednią opcję z menu kontekstowego pliku konfiguracyjnego lub narzędzia OpenVPN GUI:



Opcja ta uruchomia okno konsoli z logami programu openvpn. Utworzenie tunelu następuje po zakończeniu procedury negocjacji połączenia (patrz ostatnia linia na poniższym zrzucie ekranu):



Nowy adres IP po ustanowieniu tunelu widać oczywiście na interfejsie TAP:



🕮 Literatura

http://openvpn.net

🗒 Problemy do analizy:

- Czy możliwe jest uwierzytelnianie klienta tunelu VPN przez nazwę użytkownika i hasło?

💻 Zadania:

1. Skonfiguruj połączenie VPN przy użyciu mechanizmu *pre-shared key*.
2. Skonfiguruj połączenie VPN przy użyciu certyfikatów X.509.
3. Skonfiguruj połączenie VPN w sytuacji gdy jedna strona tunelu działa pod kontrolą systemu operacyjnego Windows.

Dodatek

Generacja certyfikatów X.509

Wykorzystany zostanie skrypt CA.pl z katalogu /usr/share/ssl/misc

1. utworzenie certyfikatu urzędu certyfikacji CA:

```
./CA.pl -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.pl -newreq
```

Należy podać wszystkie dane potrzebne do utworzenia certyfikatu dla danej strony połączenia, przy czym najważniejsze jest pole *Common Name*.

3. podpisanie klucza z pkt. 2, czyli wystawienie certyfikatu:

```
./CA.pl -sign
```

Po wykonaniu wszystkich kroków będziemy dysponować trzema niezbędnymi plikami:

cacert.pem – certyfikat CA
newkey.pem – klucz prywatny danej strony połączenia (zaszyfrowany hasłem)
newcert.pem – certyfikat z kluczem publicznym danej strony połączenia

4. Następnie należy wygenerować element pierwotny metody Diffiego-Helmana:

```
openssl dhparam -out dh.pem 2048
```

5. W razie potrzeby można też odszyfrować klucz prywatny wygenerowany wcześniej:

```
openssl rsa -in newkey.pem -out key.pem
```