Konfiguracja połączenia tunelowanego VPN

z wykorzystaniem mechanizmów oprogramowania OpenWrt oraz OpenVPN

Niniejszy samouczek opisuje krok po kroku działania mające na calu uruchomienie serwera połączenia sieci wirtualnej VPN na routerze wyposażonym w alternatywne oprogramowanie OpenWrt.

2011-4-16

VPN (*Virtual Private Network*) Wirtualna Sieć Prywatna opisywana jako tunel, przez który płynie ruch w ramach sieci prywatnej pomiędzy klientami końcowymi za pośrednictwem sieci publicznej takiej jak Internet. Sieć ta istnieje jedynie jako struktura logiczna działająca w rzeczywistości w ramach sieci publicznej. Pomimo takiego mechanizmu działania stacje końcowe mogą korzystać z VPN dokładnie tak jak gdyby istniało pomiędzy nimi fizyczne łącze prywatne.

Potrzebne narzędzia:

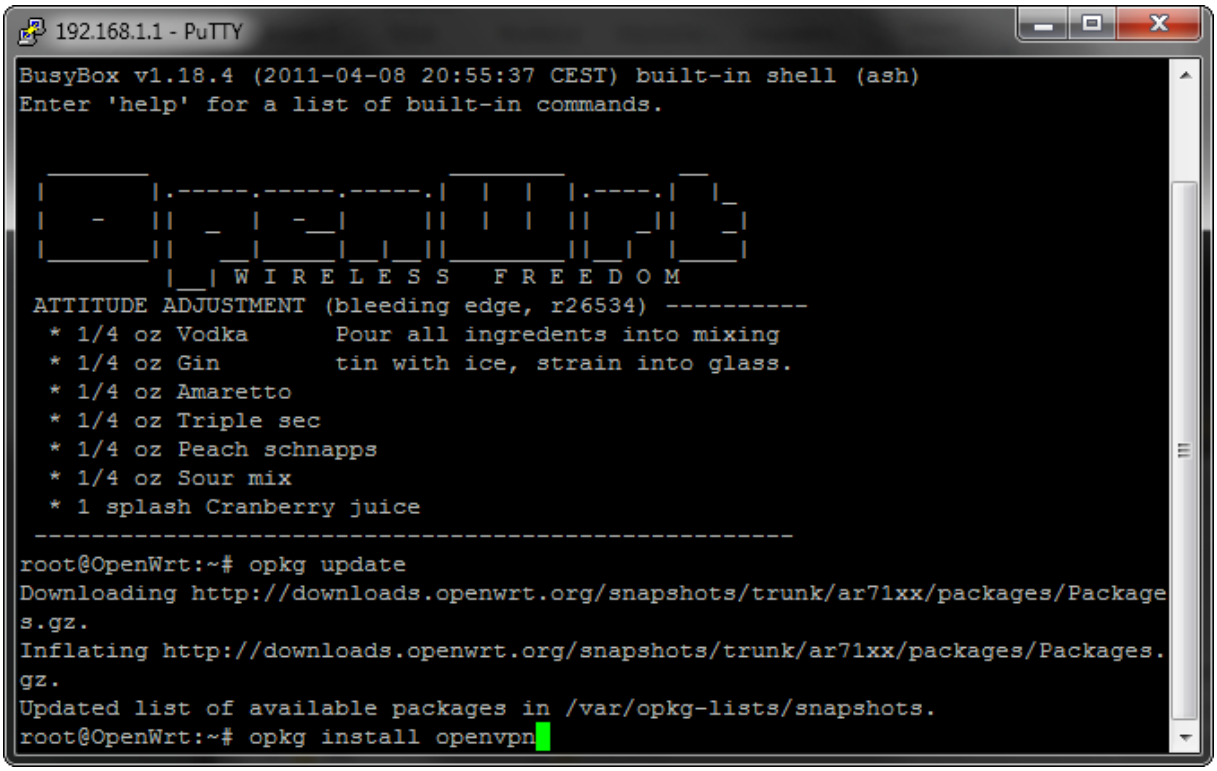
- OpenVPN dla Windows
- Router wyposażony w OpenWRT
- Putty
- WinSCP

Zainstalowano klienta OpenVPN dla Windows, a następnie z pomocą skrótu „Generate a static OpenVPN key”, wygenerowano klucz „key.txt” potrzebny do nawiązania tynelowanego połączenia.

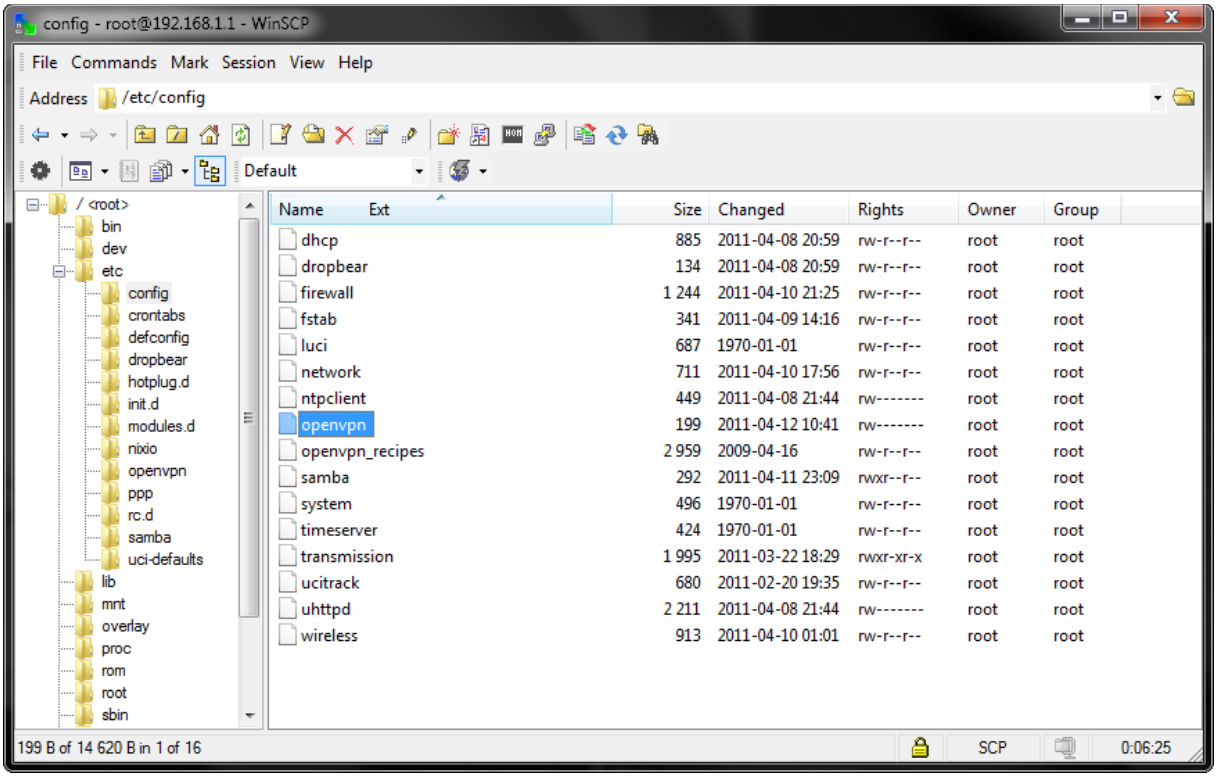
Połączono się z konsolą OpenWrt poprzez SSH w Putty. Po zalogowaniu przeprowadzono instalację OpenVPN. Wprowadzono w tym celu następujące komendy:

opkg update

opkg install openvpn

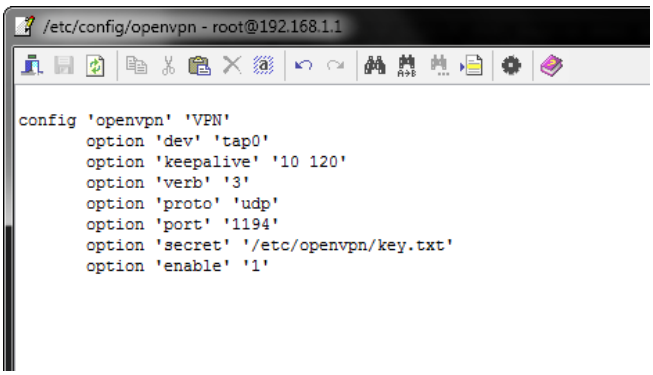


Zainstalowano WinSCP. Zalogowano z użyciem protokołu SCP oraz przeprowadzono edycję pliku „/etc/config/openvpn” …



… poprzez utworzenie następującego wpisu, stanowiącego konfigurację serwera OpenVPN o przykładowej nazwie „VPN”:

```
config 'openvpn' 'VPN'
        option 'dev' 'tap0'
        option 'keepalive' '10 120'
        option 'verb' '3'
        option 'proto' 'udp'
        option 'port' '1194'
        option 'secret' '/etc/openvpn/key.txt'
        option 'enable' '1'
```

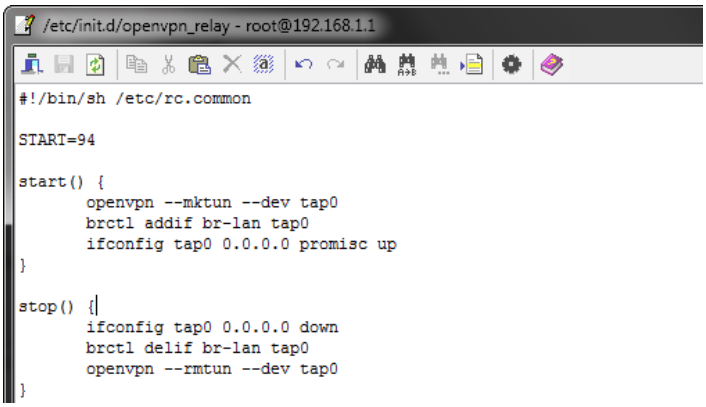


W katalogu „/etc” utworzono podfolder o nazwie „openvpn” i skopiowano do niego utworzony na początku plik „key.txt”.

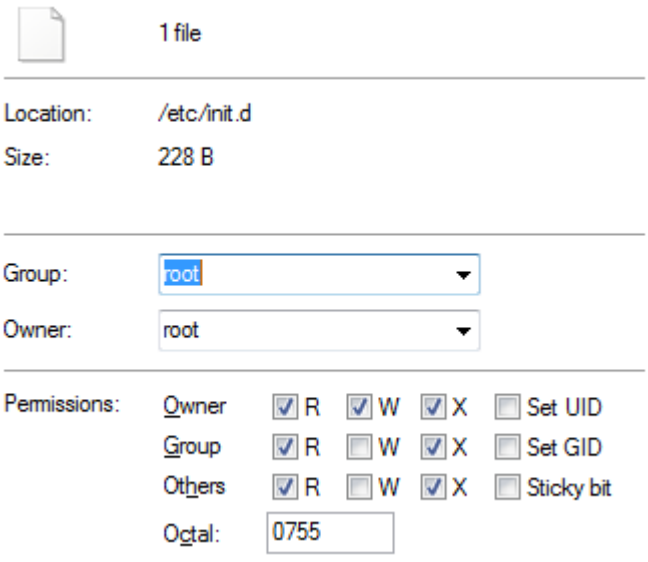
W katalogu „/init.d”, utworzono nowy plik, dla przykładu: „openvpn_relay”.

Jego treść stanowi następujący skrypt, który przed startem właściwego programu tworzy wirtualny interfejs „tap0” oraz łączy jego pulę adresową z pulą właściwego interfejsu LAN:

```
#!/bin/sh /etc/rc.common

START=94

start() {
        openvpn --mktun --dev tap0
        brctl addif br-lan tap0
        ifconfig tap0 0.0.0.0 promisc up
}

stop() {
        ifconfig tap0 0.0.0.0 down
        brctl delif br-lan tap0
        openvpn --rmtun --dev tap0
}
```



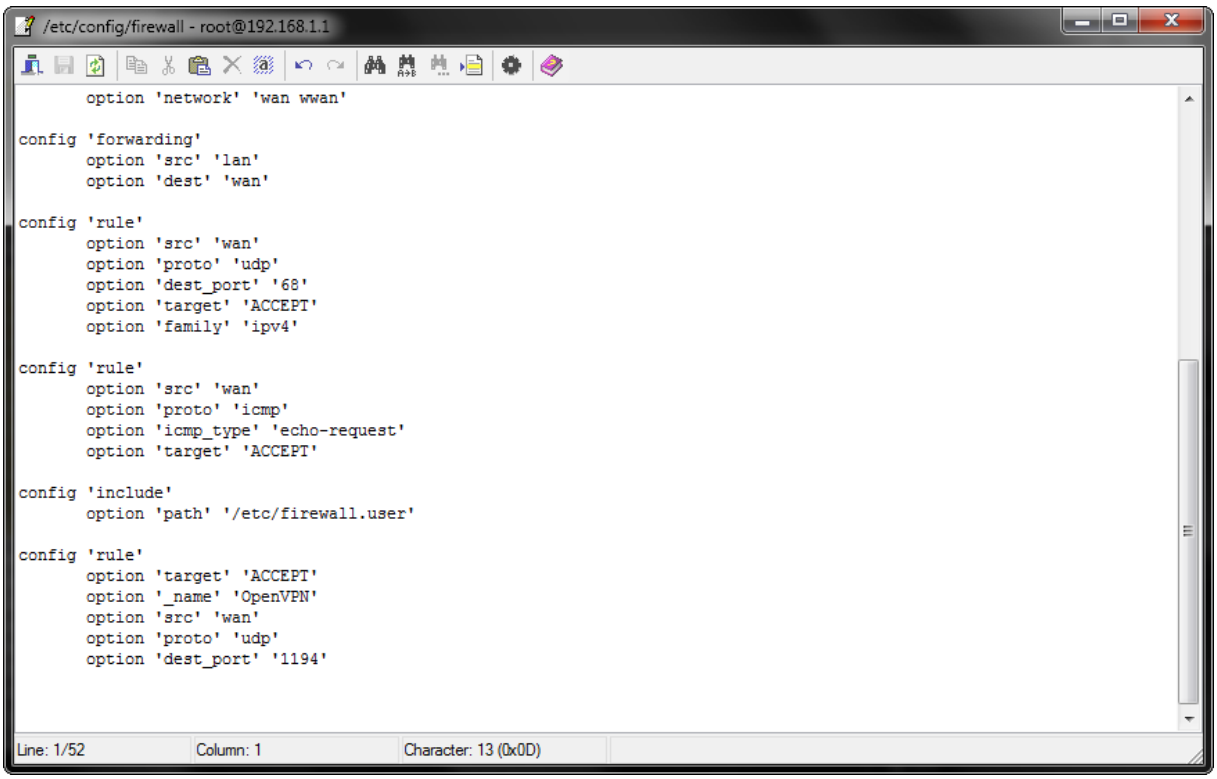
Zmieniono atrybuty nowo utworzonego pliku:



Skonfigurowano firewall poprzez stworzenie stosownych reguł pozwalających na dostęp od strony sieci WAN.

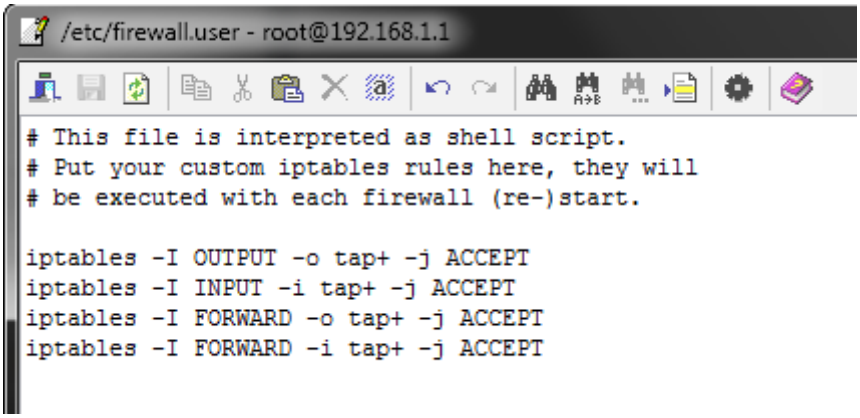
W pliku „/etc/config/firewall” dodano następującą sekcję:

```
config 'rule'
        option 'target' 'ACCEPT'
        option '_name' 'OpenVPN'
        option 'src' 'wan'
        option 'proto' 'udp'
        option 'dest_port' '1194'
```



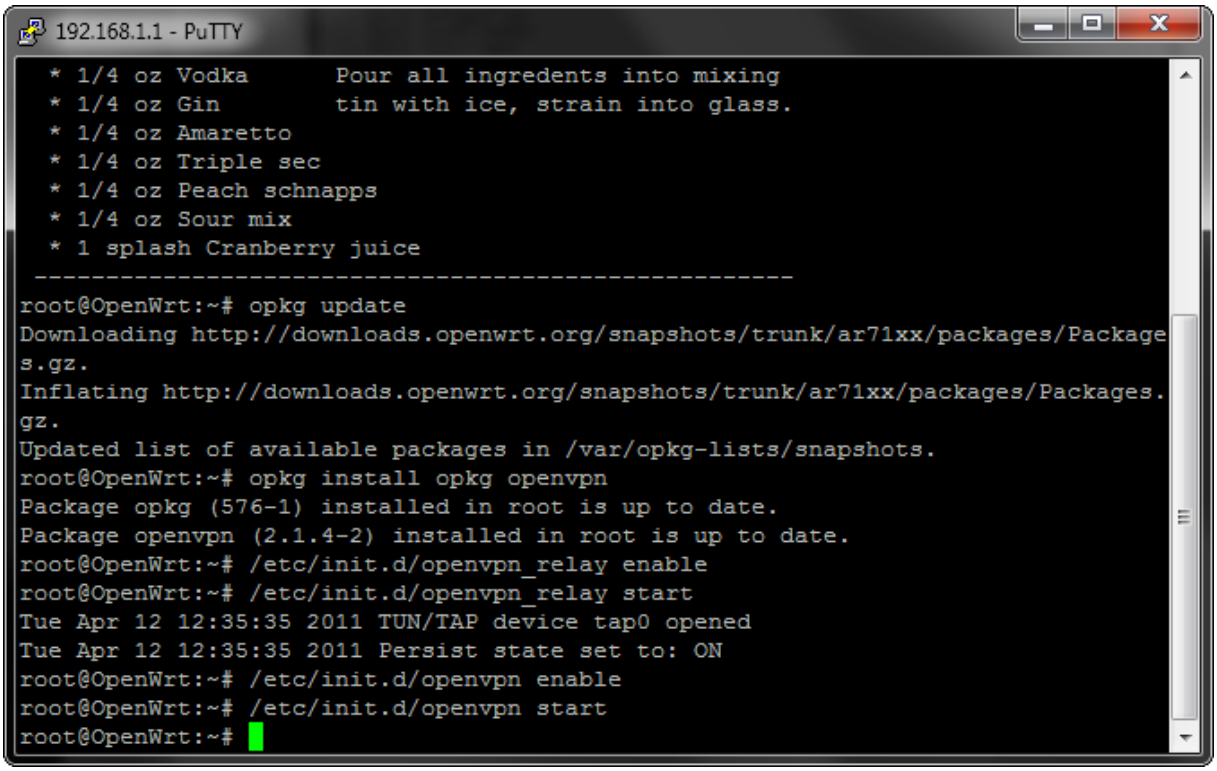
Zawartość pliku „/etc/firewall.user” wyedytowano poprzez dodanie wpisu mającego na celu stworzenie przekierowania do interfejsu „tap”:

iptables -I OUTPUT -o tap+ -j ACCEPT
iptables -I INPUT -i tap+ -j ACCEPT
iptables -I FORWARD -o tap+ -j ACCEPT
iptables -I FORWARD -i tap+ -j ACCEPT



W konsoli wprowadzono komendy mające za zadanie dodać do autostartu nowo utworzone skrypty:

/etc/init.d/openvpn_relay enable
/etc/init.d/openvpn_relay start
/etc/init.d/openvpn enable
/etc/init.d/openvpn start

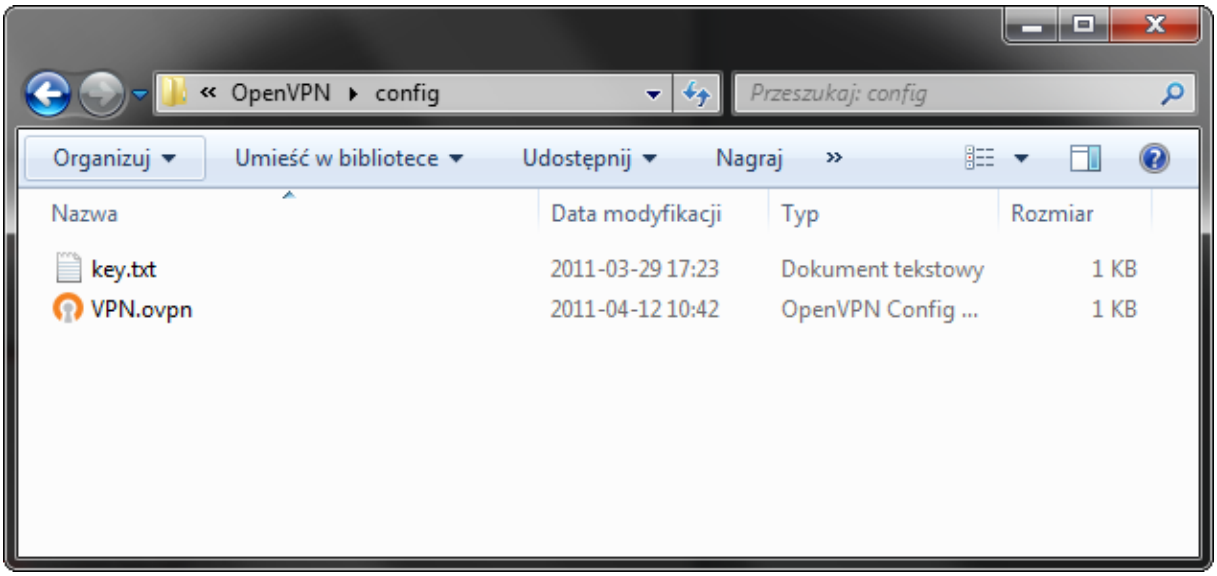


W systemie Windows stworzono konfigurację połączenia klienta OpenVPN:

W folderze:

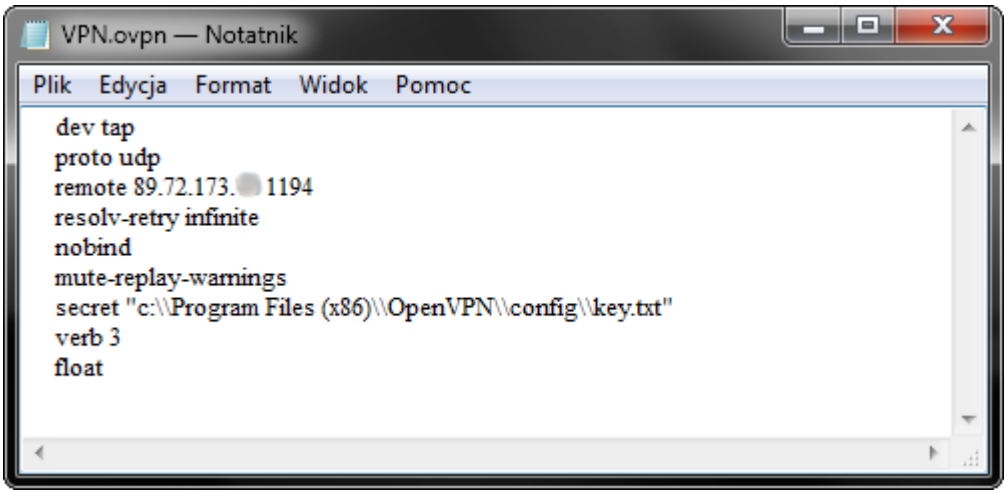
C:\Program Files\OpenVPN\config

bądź C:\Program Files (x86)\OpenVPN\config dla systemów 64 bitowych

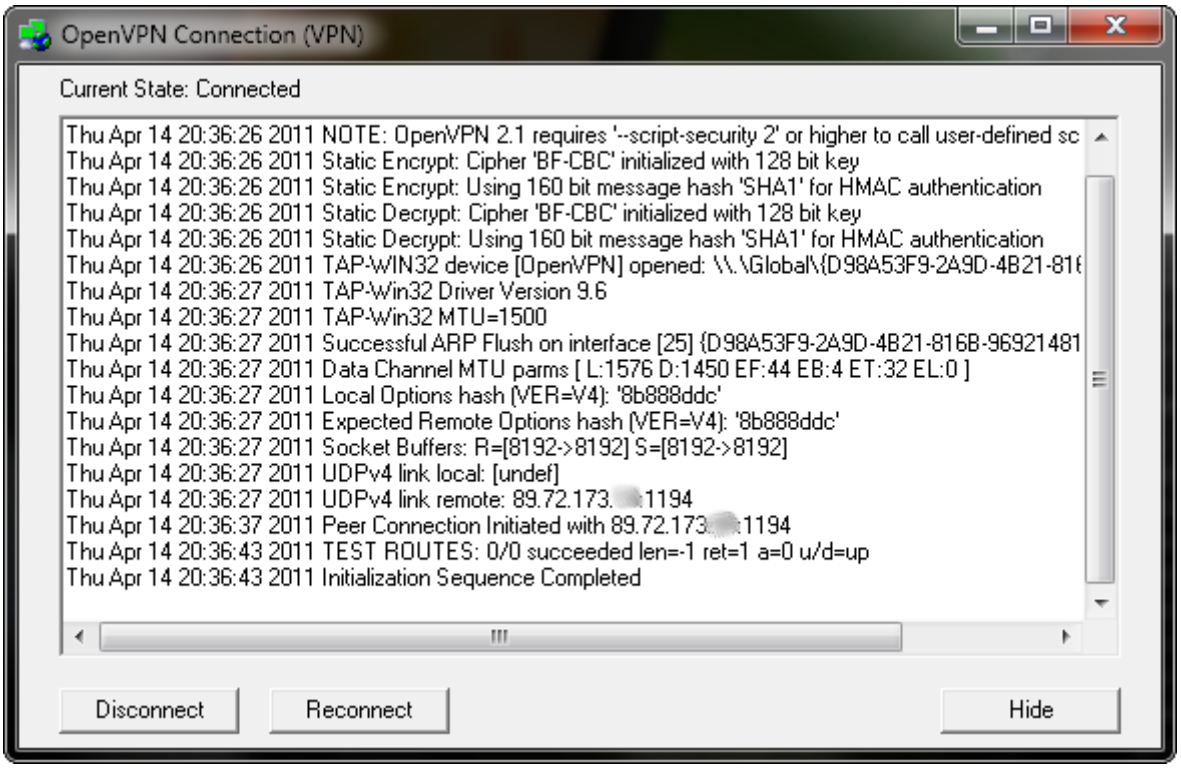


utworzono plik o nazwie przykładowej nazwie „VPN.opvn” oraz następującej treści:

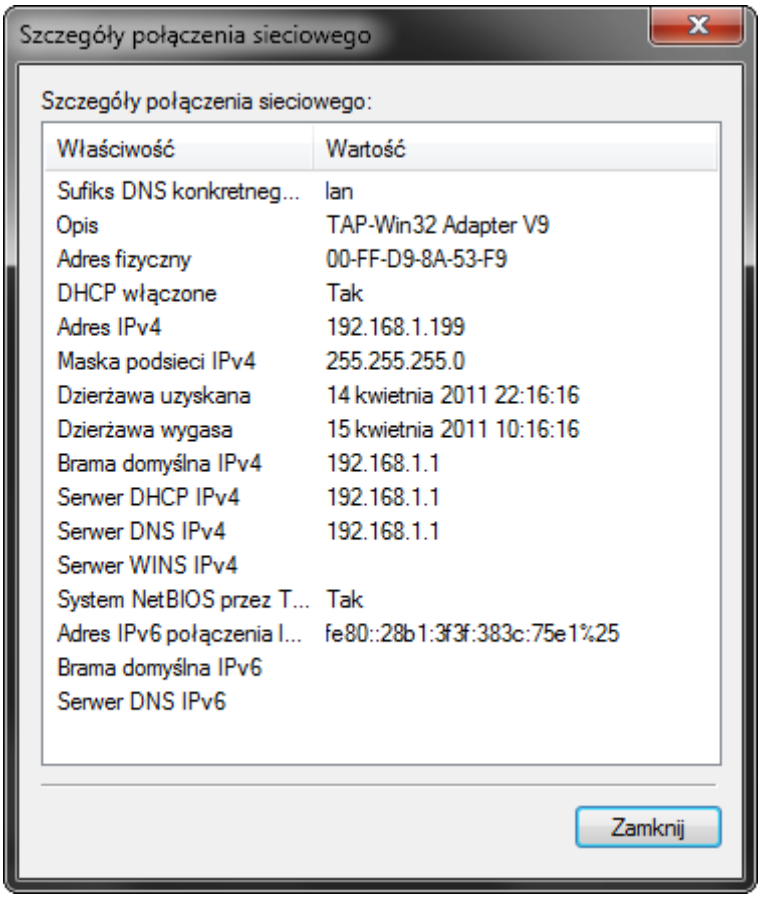
```
dev tap
proto udp
remote zewnętrzne IP 1194
resolv-retry infinite
nobind
mute-replay-warnings
secret "c:\\Program Files\\OpenVPN (x86)\\config\\key.txt”
verb 3
float
```



Uruchomiono aplikację OpenVPN GUI.



Po wykonaniu powyższych kroków wirtualnej karcie sieciowej nadany został adres IP pochodzący z lokalnej puli routera, z którym nawiązano tunelowane połączenie.



Dzięki temu połączenie ze sterownikiem zrealizowano tak jakby znajdował się on w tej samej sieci lokalnej.